AD/A-003 851

PROCEEDINGS OF THE ARMY NUMERICAL ANALYSIS CONFERENCE (11TH) HELD AT FRANKFORD ARSENAL, PHILADELPHIA, PENNSYLVANIA, ON 13-14 FEBRUARY 1974

Army Mathematics Steering Committee

Prepared for:

Army Research Office
Office of the Chief of Research, Development and Acquisition (Army)

December 1974

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ARO Report 74-2	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER: AD/A003 851	
4. TITLE (and Subtitle): PROCEEDINGS OF THE 1974 ARMY NUMERICAL ANALYSIS CONFERENCE	5. TYPE OF REPORT & PERIOD COVERED: Interim Technical Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Army Mathematics Steering Committee on behalf of the Chief of Research, Development & Acquisition	12. REPORT DATE: December 1974
	13. NUMBER OF PAGES: ~~601~~ 602
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Army Research Office, AMXRO, Box CM, Duke Station, Durham, North Carolina 27706	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited. The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

PRICES SUBJECT TO CHANGE

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20 if different from Report)

18. SUPPLEMENTARY NOTES

This is a technical report resulting from the 1974 Army Numerical Analysis Conference. It contains papers on computer aided design and engineering, as well as papers on numerical analysis.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

integer programming
target algorithm
computer modeling
munition disposal
Bayesian approach
computer graphics
numerical convergence
imbedding methods
plastic deformation
optimization methods
hybred computer solution
helium refrigeration cycles
X-ray analysis of ammunition
air defense systems
graphical firing scales
numerical integration
nonlinear analysis
dynamic response of continua
stability of a motor
quasi-Newton method
adaptive nonlinear estimation
magnetic systems
rates of convergence
mine detector design analysis
topographical data
smooth contours
plasma simulation models
acoustic bearing sensor array

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified (602)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

U. S. ARMY RESEARCH OFFICE

Report 74-2

December 1974

PROCEEDINGS OF THE 1974 ARMY NUMERICAL ANALYSIS CONFERENCE

Sponsored by the Army Mathematics Steering Committee

Host

U. S. Army Frankford Arsenal

Philadelphia, Pennsylvania

13-14 February 1974

U. S. Army Research Office
Box CM, Duke Station
Durham, North Carolina

FOREWORD

The 1974 Army Numerical Analysis Conference, sponsored by the Army Mathematics Steering Committee (AMSC), had as its host the U.S. Army Frankford Arsenal, Philadelphia, Pennsylvania, and was held on the dates of 13 and 14 February 1974. Mr. Sylvan Eisman, Chairman on Local Arrangements, shouldered the major share of the responsibility for the conduction of the conference. Those in attendance would like to thank him and other members of his committee for doing an outstanding job of arranging physical accommodations and handling the many problems that arose during the course of the meeting.

No conference in this series was held in 1973. This fact may be just one of the reasons there was such a large number of papers submitted for this eleventh meeting. The theme of this conference was "Optimal Use of Computers in Army R&D," and many of the contributed papers that were accepted for the program emphasized this topic. A new feature in this conference was a concluding meeting where each chairman presented a brief summary and a critique of the papers in his technical session.

In addition to the above-mentioned speakers, three persons gave invited addresses. The first of these was delivered by Dr. Mel Pirtle, who described the large-scale centralized computer facility at the NASA Ames Research Center located at Moffett Field, California. Professor Magnus R. Hestenes of the University of California at Los Angeles and Thomas J. Watson Research Center, Yorktown Heights, New York, gave a survey of various optimization techniques. In particular, he described the steepest descent and conjugate direction algorithms and compared their relative advantages. Dr. J. M. Yohe, the third invited speaker, discussed the computer facilities at the Mathematics Research Center at the University of Wisconsin. He also described various computer routines and packages which have been developed and are available for use by Army scientists.

Another important phase of this conference was the presentation of the citation noted below.

D E P A R T M E N T O F T H E A R M Y

CERTIFICATE OF ACHIEVEMENT

Awarded to John H. Giese

The Army Mathematics Steering Committee expresses its appreciation of and its gratitude for the valuable contribution made by Dr. John H. Giese as Chairman of the Computing and Numerical Analysis Subcommittee. His keen insight into the role of computers and numerical analysis in the business of Army research and development is best summarized by his statement to the subcommittee: "Ask not what mathematics can do for us,

but rather ask what new things it has done for us lately!" His missionary activity to bring the frontiers of computers and numerical analysis to bear on the mission of the subcommittee, on his own laboratory, and on the entire community of the Army computer users is herewith recognized with great appreciation.

The Army Mathematics Steering Committee expresses its gratitude for Dr. John H. Giese's contributions and looks forward to his continuing participation in the work of the committee and for his stimulating ideas.

12 February 1974

Lothrop Mittenthal
COL, AD
Chairman, Army Mathematics
Steering Committee

In the photograph the gentleman on the right is Colonel Lothrop Mittenthal, Commander of the Army Research Office at Durham, North Carolina. He is seen presenting the certificate to Dr. John H. Giese. As chairman of the AMSC Subcommittee on Numerical Analysis and Digital Computers, Dr. Giese organized the first ten of this series of conferences. The chairmanship job has been taken over by Dr. R. P. Uhlig of the U. S. Army Materiel Command, Alexandria, Virginia.

(U.S. ARMY PHOTOGRAPH)

TABLE OF CONTENTS*

*This Table of Contents lists only the papers that are published in this technical manual. For a list of all the papers presented at the 1974 Conference on Numerical Analysis see the copy of the Agenda.

A G E N D A

1974 ARMY NUMERICAL ANALYSIS CONFERENCE
U. S. Army Frankford Arsenal, Philadelphia, Pennsylvania

Wednesday, 13 February 1974

0815-0845 REGISTRATION - Building 12

0845-0900 OPENING OF THE CONFERENCE - Executive Conference Room Building 12A

WELCOMING REMARKS - Colonel Rex D. Wing, Commanding Officer, U.S. Army, Frankford Arsenal

0900-1000 GENERAL SESSION I - Executive Conference Room

CHAIRMAN: Dr. Ronald P. Uhlig, Chief, Scientific and and Management Information Division, Army Materiel Command, Alexandria, Va.

LARGE SCALE CENTRALIZED COMPUTER FACILITY
Dr. Mel Pirtle, Director, Institute for Advanced Computation, NASA Ames Research Center, Moffett Field, CA 94035

1000-1020 BREAK

1020-1200 TECHNICAL SESSION I - Room A

CHAIRMAN: Dr. Walter Foster, U.S. Army Surgeon General, Washington, D.C.

PROGRAM INTEGRATION FOR OPTIMAL SYSTEM DESIGN
Dr. Leonard F. Nichols and Ferdinand Scerbo, Picatinny Arsenal, Dover, NJ

NON LINEAR AND MIXED INTEGER PROGRAMMING
Bruce D. Barnett and Byron O. White, Picatinny Arsenal, Dover, NJ

DEVELOPMENT OF A TARGET ALGORITHM FOR USE WITH UNATTENDED GROUND SENSORS
Raymond Coakley, US Army Mobility Equipment Research and Development Center, Fort Belvoir, VA

1020-1200 TECHNICAL SESSION I (Continued)

EDGEWOOD ARSENAL INCENERATION PROGRAM
William R. Brankowitz and William Shulman,
Edgewood Arsenal, Aberdeen Proving Ground, MD

EDGEWOOD ARSENAL POLLUTION ABATEMENT SCRUBBER PROGRAM
William Shulman and William R. Brankowitz,
Edgewood Arsenal, Aberdeen Proving Ground, MD

1020-1200 TECHNICAL SESSION II - Executive Conference Room

CHAIRMAN: Dr. Ralph Harris, U. S. Army Management
Engineering Training Agency, Rock Island, IL

A COMPUTER MODELING TECHNIQUE APPLIED TO PRIORITY RANKING OF DEVELOPMENT PROGRAMS
Dr. Edward H. Gamble, US Army Test and Evaluation
Command, Aberdeen Proving Ground, MD

ECONOMIC, RISK AND SYSTEMS ANALYSIS OF THE CHEMICAL AGENT/MUNITION DISPOSAL SYSTEM (CAMDS)
John Seigh and Lynn Davis, Edgewood Arsenal,
Aberdeen Proving Ground, MD

FORECAST OF SCHEDULE/COST STATUS UTILIZING PERFORMANCE REPORTS OF THE COST/SCHEDULE CONTROL SYSTEMS CRITERIA: A BAYESIAN APPROACH
Dr. M. Zaki El-Sabban, US Army Aviation Systems
Command, St. Louis, MO

COMPUTER GRAPHICS APPLIED TO TEACHING OF MATHEMATICAL PRINCIPLES AT THE UNITED STATES MILITARY ACADEMY
CPT Arthur G. Bonifas, Department of Mathematics,
US Military Academy, West Point, NY 10996

1020-1200 TECHNICAL SESSION III - Room B

CHAIRMAN: Dr. John H. Giese, Chief, Applied
Mathematics Laboratory, Ballistics Research
Laboratories, Aberdeen, MD

APPROXIMATIONS IN EVALUATING THE RADIATIVE TRANSFER EQUATION
Dr. Louis D. Duncan, Atmospheric Sciences Laboratory,
US Army Electronics Command, White Sands Missile
Range, NM 88002

Wednesday AM and PM

1020-1200 TECHNICAL SESSION III (Continued)

ON THE NUMERICAL CONVERGENCE OF MATRIX EIGENVALUE PROBLEMS DUE TO CONSTRAINT CONDITIONS
Julian J. Wu, Benet Weapons Laboratory, Watervliet Arsenal, Watervliet, NY 12189

AN IMBEDDING METHOD FOR NONLINEAR MATRIX EIGENVALUE PROBLEMS OF STABILITY AND VIBRATION
E. Zagustin, R. E. Kalaba, and M. R. Scott, California State University, Long Beach, CA 90840

NUMERICAL SOLUTION SCHEMES FOR HIGHLY NONLINEAR STATIC STRUCTURAL BEHAVIOR
John F. McNamara, Structural Mechanics Branch, CERL and University of Illinois, Champaign-Urbana, IL 61801

DEVELOPMENT OF COMPUTERIZED NUMERICAL METHODS FOR APPROXIMATING THE VELOCITY AND TEMPERATURE DISTRIBUTION IN NONLINEAR, AXISYMMETRIC SOLIDS UNDERGOING LARGE PLASTIC DEFORMATION
Paul Gordon, Pitman-Dunn Laboratory, Frankford Arsenal, Philadelphia, PA and Taylan Altan, Battelle Columbus Laboratories, Columbus, OH

1200-1315 LUNCH

1315-1415 GENERAL SESSION II - Executive Conference Room

CHAIRMAN: Colonel Lothrop Mittenthal, Commanding Officer, U.S. Army Research Office, Durham

CONJUGATE DIRECTIONS METHODS IN OPTIMIZATION
Professor Magnus R. Hestenes, University of California, Los Angeles and Thomas J. Watson Research Center, Yorktown Heights, NY 10598

1425-1640 TECHNICAL SESSION IV - Room A

CHAIRMAN: Dr. Ronald P. Uhlig, Chief, Scientific and Management Information Division, Army Materiel Command, Alexandria, VA

1425-1640 TECHNICAL SESSION IV (Continued)

AN ADVANCED HYBRID COMPUTER SYSTEM FOR SIMULATION AND DATA REDUCTION
A. Gerald Edwards, Picatinny Arsenal, Dover, NJ and Aldric Saucier, Army Materiel Command, Alexandria, VA

COMPUTERGRAPHICS LANGUAGE FOR YOUR DESIGN EQUATIONS
Robert I. Isakower and Robert E. Barnas, Picatinny Arsenal, Dover, NJ

1525-1540 BREAK

HYBRID COMPUTER SOLUTION TECHNIQUES FOR LAPLACE'S EQUATION
J. Thomas Broach and Robert M. McKechnie III, US Army Mobility Equipment Research and Development Center, Fort Belvoir, VA

MINICOMPUTER VIRTUAL MEMORY TECHNIQUE FOR DATA
Dr. Larry I. Amstutz, US Army Mobility Equipment Research and Development Center, Fort Belvoir, VA

ANALYSIS PROCEDURE FOR OPTIMIZING HELIUM REFRIGERATION CYCLES
Dr. Larry I. Amstutz and Russell Eaton III, US Army Mobility Equipment Research and Development Center, Fort Belvoir, VA

COMPUTER AIDED X-RAY ANALYSIS OF SELECTED AMMUNITION MATERIALS
F. Witt, Pitman-Dunn Laboratory, Frankford Arsenal, Philadelphia, PA

1425-1640 TECHNICAL SESSION V - Executive Conference Room

CHAIRMAN: Dr. Edmund Inselmann, Office of the Chief Mathematician, Army Materiel Command, Alexandria, VA

MODULAR FORCE PLANNING SYSTEM (MFPS)
CONSTRAINED FORCE MODEL (CONFORM)
E. Pederson and S. Dix, USA Management Systems Support Agency, Operations Research Branch, Systems Development Division, Washington, D.C. 20310

1425-1640 TECHNICAL SESSION V (Continued)

THREE DIMENSIONAL AIR DEFENSE KINEMATIC LAUNCH AND INTERCEPT BOUNDARY COMPUTER PROGRAM
J. L. Harris, Aeroballistics Directorate, US Army Missile RD&E Laboratory, US Army Missile Command, Redstone Arsenal, AL

1525-1540 BREAK

A BACKWARD SOLUTION COMPUTING MISS DISTANCE RESULTING FROM INPUT ERRORS TO GUN AIR DEFENSE SYSTEMS
T. H. Slook, Fire Control Development and Engineering Directorate, Frankford Arsenal, Philadelphia, PA

ANALYSIS OF CLOSED LOOP FIRE CONTROL SYSTEMS FOR TANKS
Louis R. Cerrato and Kenneth R. Pfleger, Fire Control Development and Engineering Directorate, Frankford Arsenal, Philadelphia, PA

MATHEMATICAL MODEL FOR PREDICTION OF VISUAL RANGES ATTAINABLE WITH OPTICAL SIGHTS
David L. Steinberg and Wright H. Seidmore, Fire Control Development and Engineering Directorate, Frankford Arsenal, Philadelphia, PA

COMPUTER AIDED DESIGN OF GRAPHICAL FIRING SCALES
Diana T. Dadamo and Joseph W. Kaszupski, Fire Control Development and Engineering Directorate, Frankford Arsenal, Philadelphia, PA

1425-1640 TECHNICAL SESSION VI - Room B

CHAIRMAN: Professor Carl De Boor, Mathematics Research Center, University of Wisconsin, Madison, WI

A COMPUTATIONAL SYSTEM FOR NUMERICAL INTEGRATION WITH RIGOROUS ERROR ESTIMATION
Professors Louis B. Rall and Julia H. Gray, Mathematics Research Center, University of Wisconsin, Madison, WI 53706

TOTALLY CONSERVATIVE METHODS FOR THE NUMERICAL INTEGRATION OF EQUATIONS OF MOTION
Dr. Robert A. La Budde, Mathematics Research Center and Professor Donald Greenspan, Computer Sciences Department and Academic Computing Center, University of Wisconsin, Madison, WI 53706

1425-1640 TECHNICAL SESSION VI (Continued)

1525-1540 BREAK

ON THE EFFECTIVE USE OF A LARGE COMPUTER PROGRAM FOR STRUCTURAL CALCULATIONS
E. Cuthill and P. Matula, Naval Ship Research and Development Center, Bethesda, MD 20034

APPLICATION OF NONLINEAR ANALYSIS (PLASTIC) TO NASTRAN USING RING ELEMENTS INCLUDING ASPECT RATIO EFFECTS
Diana L. Frederick, Munitions Development and Engineering Directorate, Frankford Arsenal, Philadelphia, PA

AN OPTIMAL COMPUTERIZED ALGORITHM FOR CALCULATING THE DYNAMIC RESPONSE OF CONTINUA
Paul F. Gordon, Pitman-Dunn Laboratory, Frankford Arsenal, Philadelphia, PA

HYDRODYNAMIC COMPUTER CODE SOLUTION OF EXPLOSIVE EXCAVATION DESIGN PROBLEMS
J. E. Lattery, Explosive Excavation Research Laboratory, US Army Corps of Engineers Waterways Experiment Station, Vicksburg, MS

****

Thursday, 14 February 1974

0800 Bus Leaves Sheraton Motor Inn for the Conference.

0830-0930 GENERAL SESSION III - Executive Conference Room

CHAIRMAN: Dr. Sylvan Eisman, Pitman-Dunn Laboratory, U.S. Army Frankford Arsenal

COMPUTER SOFTWARE DEVELOPMENT AT MRC
Professor J. M. Yohe, Assistant Director, Mathematics Research Center, University of Wisconsin, Madison, WI 53706

0930-0945 BREAK

0945-1215 TECHNICAL SESSION VII - Room B

CHAIRMAN: Professor Louis B. Rall, Mathematics Research Center, University of Wisconsin, Madison, WI 53706

0945-1215 TECHNICAL SESSION VII (Continued)

COMPUTER MODELING IN DETERMINING STABILITY OF A MORTAR REPOSITIONING NONLINEAR CONTROL SYSTEM
C. N. Shen and G. N. Woods, Benet Weapons Laboratory, US Army Watervliet Arsenal, Watervliet, NY 12189

CONVERGENCE PROPERTIES OF QUASI-NEWTON METHODS WITH APPROXIMATE LINE SEARCHES
Professor Melanie L. Lenard, Mathematics Research Center, University of Wisconsin, Madison, WI 53706

ADAPTIVE NONLINEAR ESTIMATION APPLICATION FOR TEMPERATURE FORECASTING
Newton B. Penrose, Department of Electrical Engineering, US Military Academy, West Point, NY

A METHOD TO ANALYZE NONLINEAR MAGNETIC SYSTEMS
Robert H. Haveson, Picatinny Arsenal, Dover, NJ

RATES OF CONVERGENCE FOR A CLASS OF NONLINEAR PROGRAMMING ALGORITHMS
Professor Stephen M. Robinson, Mathematics Research Center, University of Wisconsin, Madison, WI

0945-1215 TECHNICAL SESSION VIII - Room A

CHAIRMAN: Dr. William J. Sacco, Chief Applied Mathematics and Statistics Group, Biomedical Laboratory, Edgewood Arsenal, MD

MODERN LENS DESIGN ON LARGE SCALE COMPUTERS
James W. Shean, Fire Control Development and Engineering Directorate, Frankford Arsenal, Philadelphia, PA

DIAGNOSTIC FUNCTIONS FOR SEARCH AND RETRIEVAL FROM SPECTRAL DATA BANKS
D. H. Robertson, C. Merritt, Jr., US Army Natick Laboratories, Natick, MA

MATHEMATICAL MODELING AND SIMULATION OF THE CELLULOSE/TV-CELLULASE HYDROLYSIS
Chul Kim, US Army Natick Laboratories, Natick, MA

0945-1215 TECHNICAL SESSION VIII (Continued)

GEOMETRY OF FOOD PREFERENCES
T. J. Reed, H. R. Moskowitz, US Army Natick Laboratories, Natick, MA

COMPUTED ENERGY DISTRIBUTIONS OF DOUBLE-SCATTERED PHOTONS OBTAINED FOR PURPOSES OF MINE DETECTOR DESIGN ANALYSIS
Fredrick L. Roder, Douglas G. Conley, US Army Mobility Equipment Research and Development Center, Fort Belvoir, VA

COMPUTERIZED EQUIVALENT CIRCUIT MODELS OF FLUID CAPILLARIES
Joseph M. Iseman, Harry Diamond Laboratories, Washington, D.C. 20438

0945-1215 TECHNICAL SESSION IX - Executive Conference Room

CHAIRMAN: Dr. Lawrence A. Gambino, Director, Computer Science Laboratory, Engineering Topographic Laboratories, Fort Belvoir, VA

AUTOMATED PROCEDURES FOR ACQUISITION, STORAGE, MANIPULATION, AND ANALYSIS OF TOPOGRAPHIC DATA FOR USE IN SYSTEMS ANALYSIS PROBLEMS
Phillip L. Doiron, Sr., US Army Engineer Waterways Experiment Station, Vicksburg, MS

OPTIMAL REPRESENTATION OF GEOGRAPHICAL MAPS FOR COMPUTERS
Dr. Theodosios Pavlidis, Department of Electrical, Engineering, Princeton University, Princeton, NJ and Fire Control Development and Engineering Directorate, Frankford Arsenal, Philadelphia, PA

COMPUTATION OF SMOOTH CONTOURS FROM NON-UNIFORM DATA
Richard J. Blair, Benet Weapons Laboratory, Watervliet Arsenal, Watervliet, NY

THE TWO-STREAM INSTABILITY STUDIED WITH FOUR ONE-DIMENSIONAL PLASMA SIMULATION MODELS
David I. Brown, Fire Control Development and Engineering Directorate, Frankford Arsenal, Philadelphia, PA

Thursday AM and PM

0945-1215 TECHNICAL SESSION IX (Continued)

COMPUTER SIMULATION OF SYMPATHETIC DETONATION
James D. Wood, US Army Edgewood Arsenal, Aberdeen Proving Ground, MD

A PSEUDO CALIBRATION PROCEDURE FOR AN ACOUSTIC BEARING SENSOR ARRAY
Kenneth J. Dean, US Army Mobility Equipment Research and Development Center, Fort Belvoir, VA

MONTE CARLO CALCULATION OF LASER INDUCED CHANGE IN OPTICAL DENSITY
R. W. Anderson, Jr., R. E. Salomon, L. E. Harris, J. J. Mikula, Pitman-Dunn Laboratory, Frankford Arsenal, Philadelphia, PA

1215-1330 LUNCH

1330-1500 GENERAL SESSION IV - Executive Conference Room

CHAIRMAN: Dr. Sidney Ross, Technical Director, U.S. Army Frankford Arsenal

SUMMARY AND REVIEW OF TECHNICAL SESSIONS

Dr. Walter Foster	-US Army Surgeon General
Dr. Ralph Harris	-AMETA
Dr. John H. Giese	-Ballistics Research Laboratories
Dr. Ronald P. Uhlig	-AMC HQ
Dr. Edmund Inselmann	-AMC HQ
Prof. Carl de Boor	-Mathematics Research Center
Prof. Louis B. Rall	-Mathematics Research Center
Dr. William J. Sacco	-Biomedical Laboratory
Dr. Lawrence A. Gambino	-Engineering Topographic Laboratories

PROGRAM INTEGRATION FOR OPTIMAL SYSTEM DESIGN

Leonard F. Nichols
Ferdinand A. Scerbo
Concepts and Effectiveness Division
Nuclear Development and Engineering Directorate
U.S. Army Armaments Command
Picatinny Arsenal, Dover, N.J.

ABSTRACT. A generalized method for combining existing computer programs under the control of an executive program is presented. The system has been applied to integrate computer programs used in the design of projectile ammunition. The system is capable of defining key variables which may be modified under executive program control. The over-all system makes extensive use of permanent files to handle both data base and program storage as well as data analysis and optimization. Computations include interior and exterior ballistics, static shell property calculations, aerodynamic properties generation and lethal area effectiveness. The system can be operated in an interactive teletype or batch mode.

1. INTRODUCTION. Vugraph 1 - The purpose of this talk is to present progress associated with work related to the Army Materiel Command sponsored CAD-E Program titled: Integrated Projectile Systems Synthesis Model (IPSSM). I intend to discuss how the program operates, the applications programs it uses and how the data management aspects of the model are handled. I will also discuss tasks that have been accomplished since the program started and what we hope to accomplish in FY75. I will complete the talk by discussing some of the benefits that can be derived from such a system after full implementation is accomplished. The objective of the Integrated Projectile Systems Synthesis Model is to develop a complete computer model for use in the preliminary design of large caliber projectiles.

2. BACKGROUND. Vugraph 2 - In May 1970, AMC initiated a feasibility study directed toward weapon system computer modeling which would provide preliminary design parameters for tactical missile weapon system concepts. After AMC had presented the concept and preliminary comments of subordinate commands to the CAD-E Council in October 1970, the IWSSM (Integrated Weapon System Synthesis Model) Ad Hoc Working Group of the Council was established to study the concept in more detail. After a six month effort the working group concluded that such a system was both feasible and desirable but initially it should be limited to the construction of a number of computer models, each addressing a particular military commodity. It was considered too large an undertaking to develop one model which would handle all commodity designs such as guns, projectiles, missiles, aircraft, vehicles, etc.

The recommendation of the IWSSM Working Group was that each command submit a Program Data Sheet outlining a proposed activity directed toward a specific commodity. The Integrated Projectile System Synthesis Model (IPSSM) program was established at Picatinny-ARMCOM to address the preliminary design of large caliber projectiles such as artillery and mortar rounds. The Picatinny IPSSM program was funded as a CAD-E task in March 1972.

3. DESCRIPTION OF COMPUTER SYSTEM. Vugraph 3 - describes the major characteristics of IPSSM. As shown, this system is being developed for use in large caliber shell design. The major effort is in the establishment of the executive computer program which controls the execution of application programs already developed. In addition the executive program provides the data management necessary to manipulate information into and out of programs for use in other programs within the system. Graphics should play a major role in the final IPSSM model. However, at this time we are primarily working on an interactive teletype version of IPSSM. The executive program is also capable of running in batch mode. The basic guideline in the IPSSM development is to provide a useful and simple tool for the projectile design engineer. Use of this system will allow him time to analyze and compare many design innovations without the need for laborious computer set-ups and hand computations which are common in today's environment.

Vugraph 4 - shows an overall flow diagram of the IPSSM system. This model is a set of computer programs and subroutines integrated by the executive program in such a manner as to provide a realistic, interactive, computational tool for engineers and designers engaged in the development of preliminary design information for projectiles. The model is designed to perform all the calculations necessary to formulate projectile design concepts. The present IPSSM system can execute five applications programs from the batch or TTY mode. These programs perform the following computations: (1) Static Properties Calculations (2) Generation of Aerodynamic Coefficients (3) Interior Ballistics (4) Exterior Ballistics and (5) Lethal Area Computations.

The IPSSM executive program also contains options for examining and modifying common data base information. The system also permits data base changes to facilitate the execution of parametric analyses where selected variables are "linked" so that a restricted set of combination runs can be made. Special output can be stored automatically for later use.

Vugraph 5 - Lists the significant accomplishments to date. The flow diagram shown on the previous vugraph was established and existing application programs were selected, tested and stored. Some programs were modified slightly to establish input and output interfaces. The initial executive program has been developed to operate in the teletype and batch modes.

Discussions with users have been made during the past six months and the system is now available for trial use. The next vugraph describes how the executive program is used.

Vugraph 6 - The user begins by calling the executive program from the teletype. Once the user has attached the executive program, he selects the program and data to be used. The executive program will then automatically generate job control cards and data in the correct format to run the program selected. The mode of operation is then switched to batch. Following successful execution, the output data is stored in a specified data set for subsequent use or printed out on the users batch terminal. Selected lines of output may also be viewed on the teletype during the next run.

The executive program can also be used directly in the batch mode. In this case, card input is used to modify key variable data and tables.

Vugraph 7 - Describes the initial data file generation required prior to running the executive program. This is accomplished by an independent program called IPSDATA which uses input from an initial master file containing all available information pertinent to a particular initial shell design. This program sorts key variable data and generates permanent files which can later be called by executive program functions.

Vugraph 8 - Describes the input-output interface that is created by the executive program when a particular applications program is called for. The executive program can examine and modify stored data prior to executing the applications program. It also provides for the output options shown.

The next series of vugraphs provide a general discussion of applications programs presently incorporated into the IPSSM System.

Vugraphs 9 and 10 - Describe the weight program which calculates static properties of a shell given the geometry of body items and their density. The program can also be used to compute whether or not the proposed shell is stable.

Vugraphs 11 and 12 - Describe the use of the spinner program. This program calculates aerodynamic coefficients as a function of Mach number and also is capable of performing a stability analysis.

Vugraphs 13 and 14 - Illustrate the essential features of the exterior ballistics program which is used to provide trajectory information as a function of time. It also is used in conjunction with a range error analysis which I will describe in a later vugraph.

Vugraph 15 - Shows the data management scheme used to modify original data base information in such a way that the output from one program can generate input to a second program. For example, the spinner program which generates drag coefficients as a function of Mach number can be automatically transferred as input to the terminal ballistics program which I described in a previous vugraph. Modifications can be made on a permanent basis to the existing files or output data can be transferred to establish a new data base file.

Vugraph 16 - Describes the interior ballistics program currently incorporated within the IPSSM System. It is based on simplified equations which require only the items listed under input to have it function. It can be used to calculate muzzle velocity, propellant weight or maximum pressure depending on the option selected.

Vugraph 17 and 18 - Describe the major features of the lethal area program which is used to calculate the anti-personnel effectiveness of fragmenting ammunition. Vugraph 18 lists all of the features available when using this program. It is a tri-service standard program for performing these computations. Vugraph 18 shows the input and output requirements and options capabilities.

Vugraph 19 - Describes the error analysis which is controlled by the executive program and interfaces with the exterior ballistics program. Automatic operation for three selected variables is currently operational. This scheme illustrates the type of data analyses that can be incorporated into the IPSSM system. Multiple cases can be run with the executive program in an easy manner because of the way in which the key variable data is stored. Six values may be stored for each variable. Another feature of the executive program is that it can "link" key variables together so that if one variable changes value, all other variables linked to it will also change value. This feature is useful in setting up run combinations where all combinations are not required.

Vugraph 20 - Lists the tasks planned for FY75. Each item listed here can be accomplished by extending techniques described earlier. Although it appears that no technical problems exist extensive effort is required to implement these features. Training, maintenance and documentation of the system are activities which require continued emphasis. The major task during FY75 will be to provide interactive graphics capability to the maximum extent possible.

4. CONCLUSIONS. Vugraph 21 - Lists the benefits of the IPSSM program. All of these are generally self-explanatory in that such a system emphasizes the use of analytical techniques and minimizes the need for trial-and-error approaches. It also provides for the use of well documented and tested

computer programs applicable to the projectile design process. These programs are easily accessible and can be used to perform extensive parametric investigations. Data management routines provided by the IPSSM system also gives the design engineer the capability to readily examine, modify and store design information for specific projectile designs. Thus, engineering time can be significantly reduced in performing projectile design calculations.

INTEGRATED PROJECTILE SYSTEMS SYNTHESIS MODEL (IPSSM)

ARMCOM CAD-E PROJECT

PICATINNY ARSENAL
DOVER, NEW JERSEY

INTEGRATED PROJECTILE SYSTEMS SYNTHESIS MODEL (IPSSM)

BACKGROUND

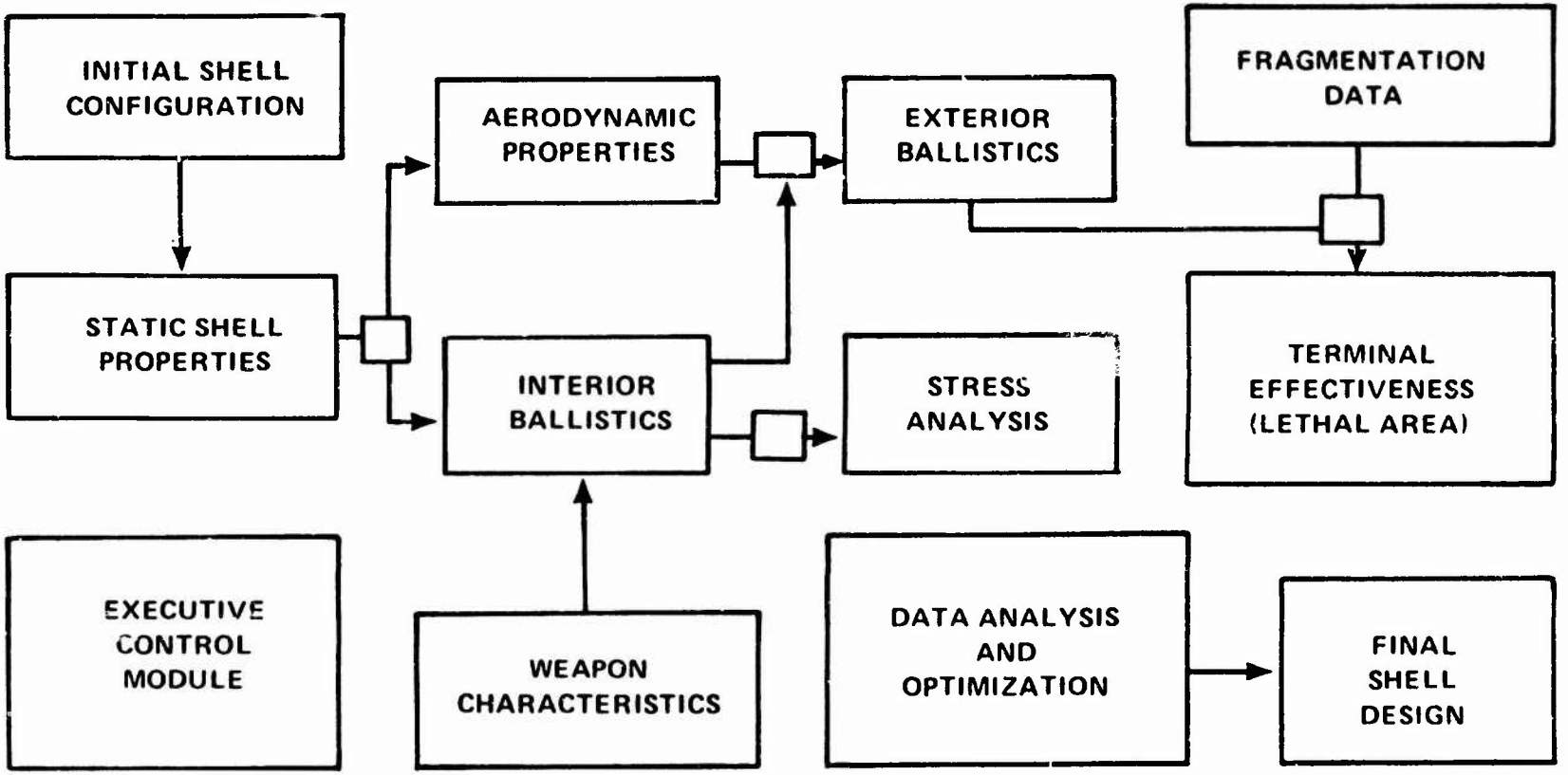
- IWSSM PROPOSAL STUDIED BY AMC
- AD HOC IWSSM WORKING GROUP ESTABLISHED
- CONCLUSION AND RECOMMENDATIONS TO CAD E COUNCIL
  - CONCEPT SHOULD BE PURSUED
  - IMPLEMENT BY MAJOR COMMODITY
  - PROPOSE IXSSM'S FOR CAD E FUNDING
- IPSSM FUNDED IN MARCH 72

MAJOR CHARACTERISTICS OF IPSSM

- LARGE CALIBER PROJECTILE DESIGNS
- EXECUTIVE MODULE TO INTEGRATE COMPUTER PROGRAM MODULES
- MAXIMUM USE OF INTERACTIVE GRAPHICS
- OVER-ALL CONTROL OF MODEL ACCOMPLISHED BY USER OPTIONS
- DESIGNS ARE EVALUATED IN TERMS OF MISSION PERFORMANCE

Vugraph 3

IPSSM FLOW DIAGRAM
(HE VERSION)



Vugraph 4

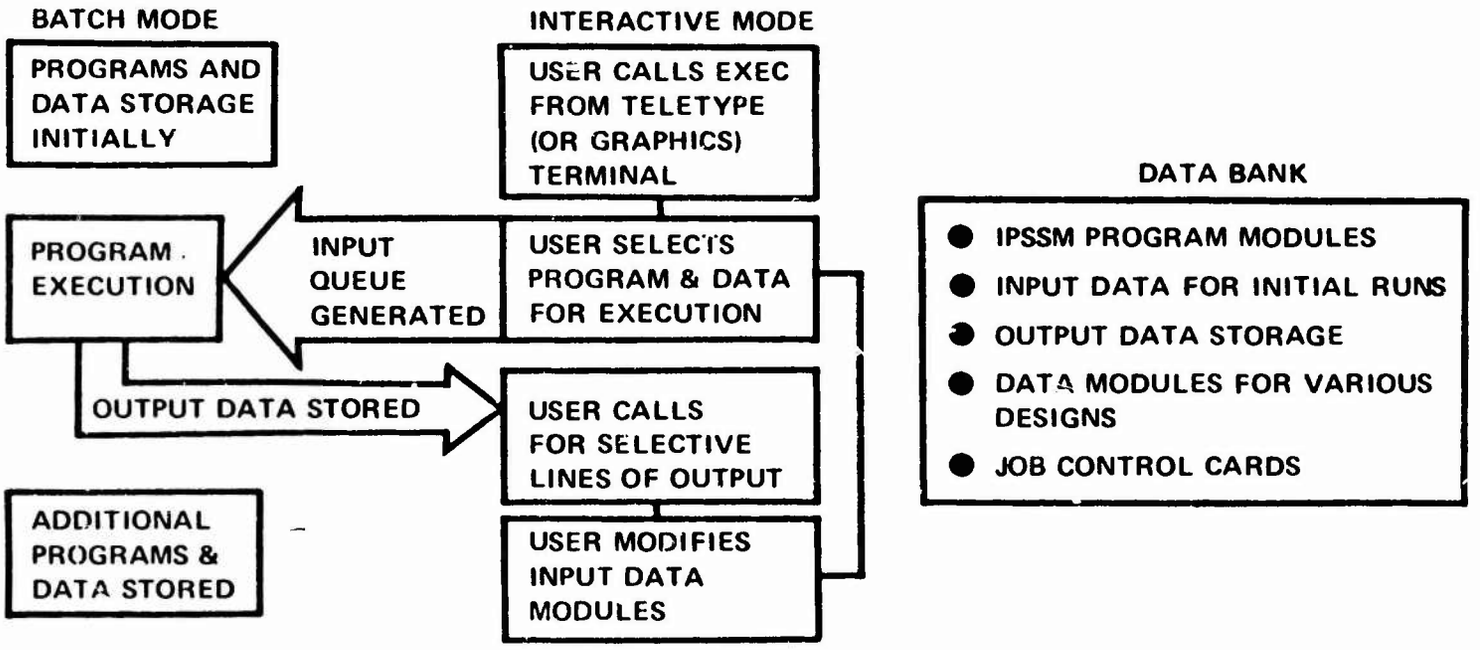
IPSSM MILESTONES

ACCOMPLISHMENTS TO DATE

- ESTABLISHED OVERALL FLOW DIAGRAM
- SURVEYED AND SELECTED EXISTING COMPUTER PROGRAMS
- TESTED AND STORED ALL PROGRAMS IN UPDATE FORMAT
- ESTABLISHED INPUT OUTPUT INTERFACE
- DEVELOPED INITIAL EXECUTIVE PROGRAM FOR TTY AND BATCH MODES
- CONDUCTED DISCUSSIONS WITH USERS
- SYSTEM AVAILABLE FOR TRIAL USE

Vugraph 5

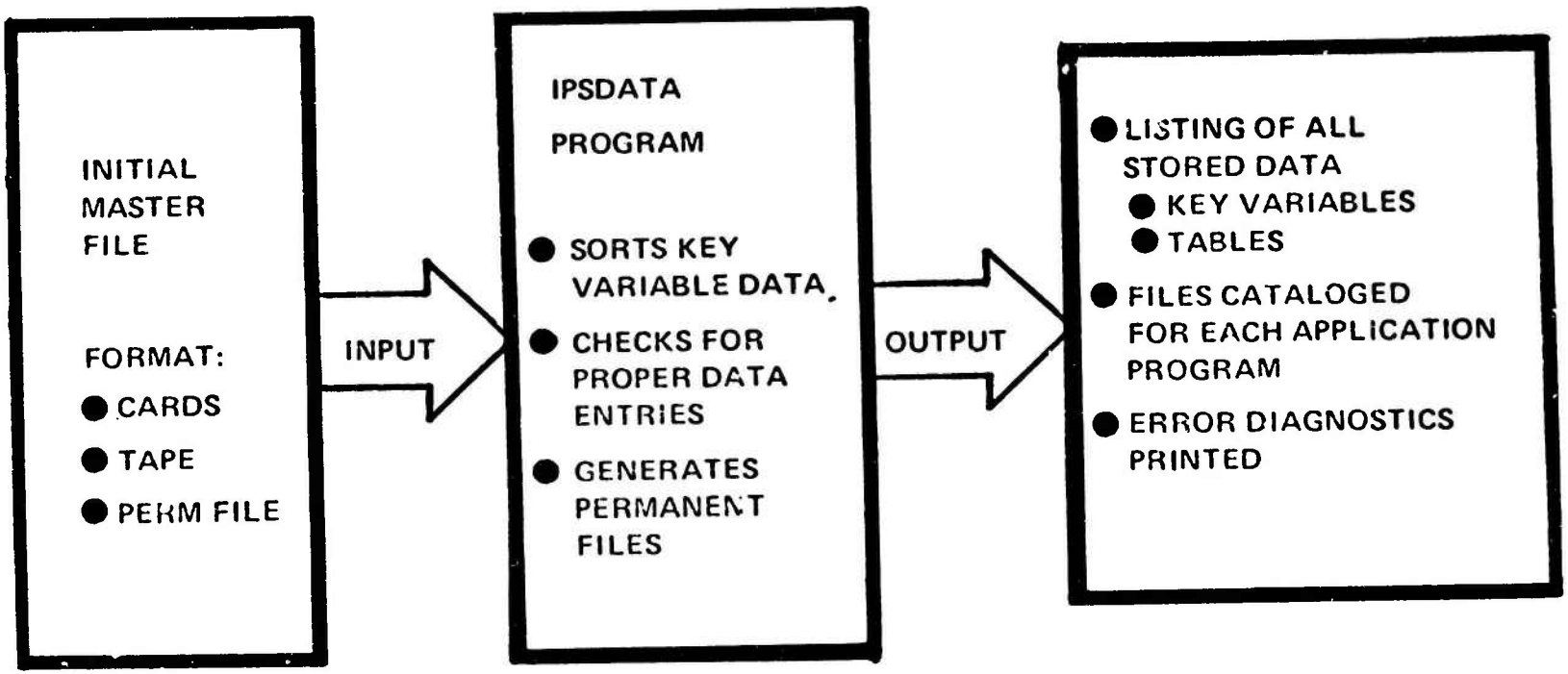
EXECUTIVE MODULE OPERATION (IPSSM)



Vugraph 6

INITIAL DATA FILE GENERATION

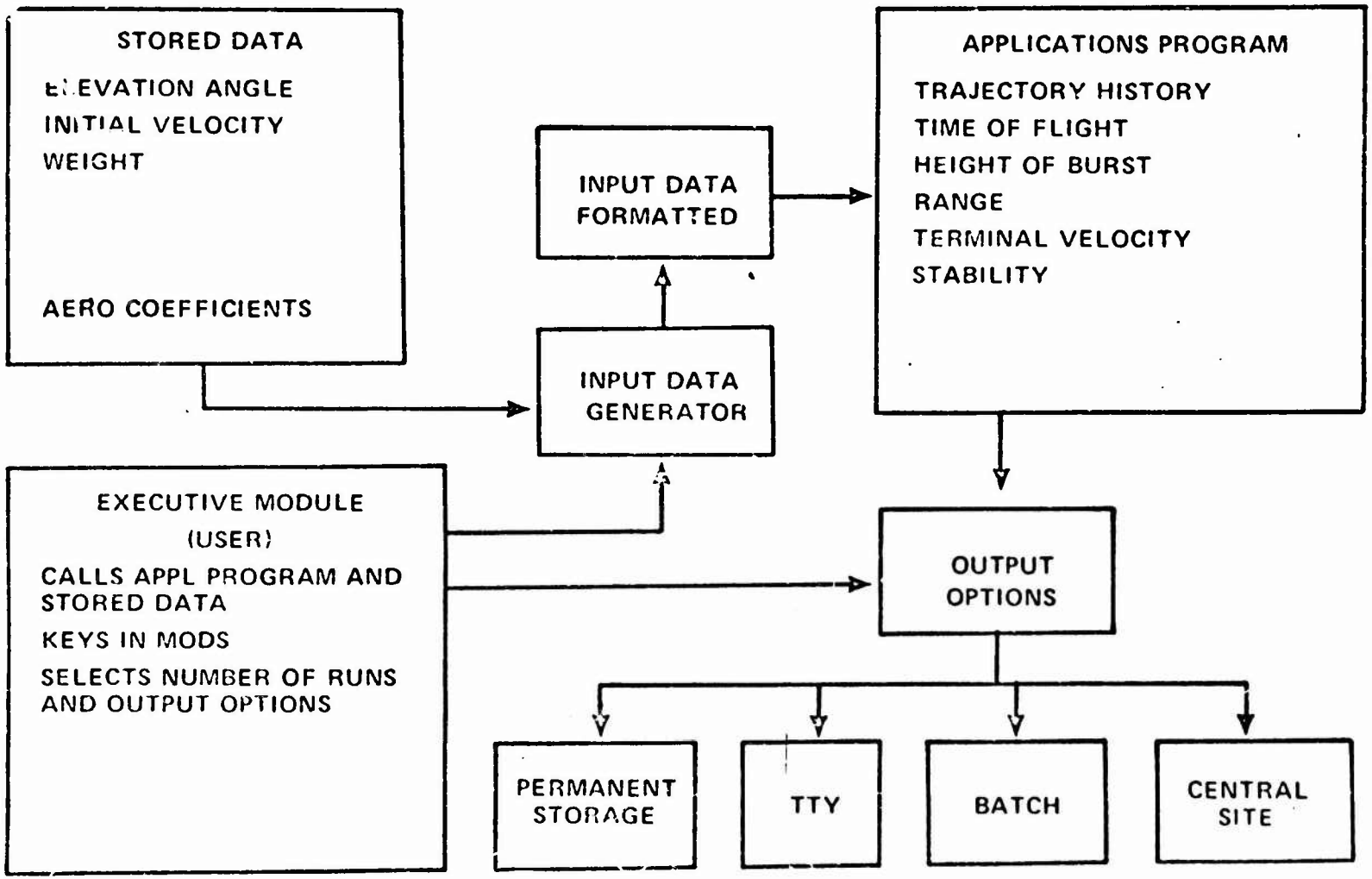
(IPSSM)



Vugraph 7

INPUT - OUTPUT INTERFACE

(EXTERIOR BALLISTICS)



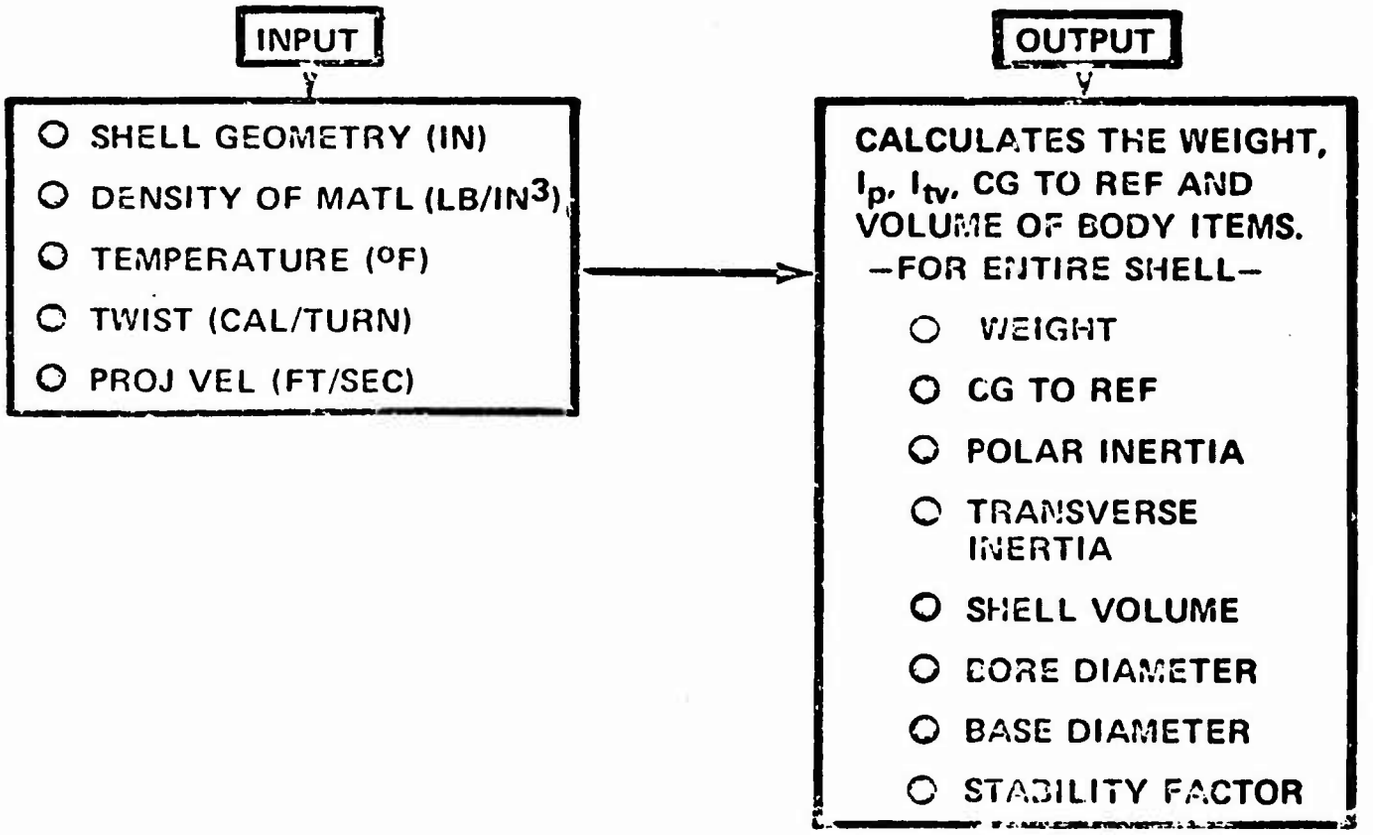
Viewgraph 6

WEIGHT COMPUTER CODE

CALCULATES PROPERTIES OF INDIVIDUAL BODY ITEMS, ENTIRE SHELL AND HAS PLOT & GRAPHIC CAPABILITIES. ALSO CALCULATES STABILITY FACTOR

Vugraph 9

WEIGHT (INPUT-OUTPUT)



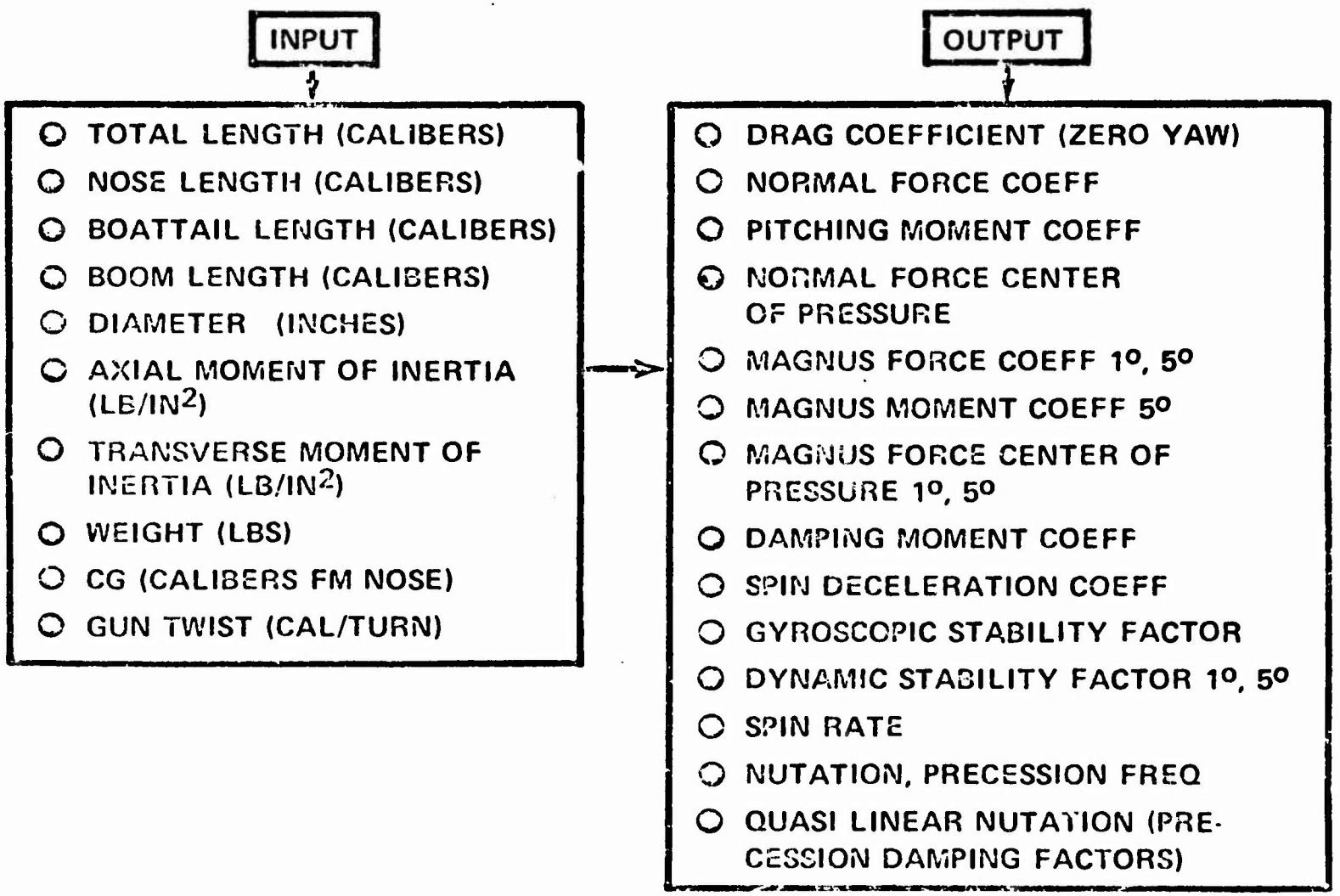
Vugraph 10

SPINNER COMPUTER CODE

CALCULATES AERODYNAMIC COEFFICIENTS FOR
VARIOUS MACH NUMBERS AND STABILITY ANALYSIS

Vugraph 11

SPINNER (INPUT-OUTPUT)



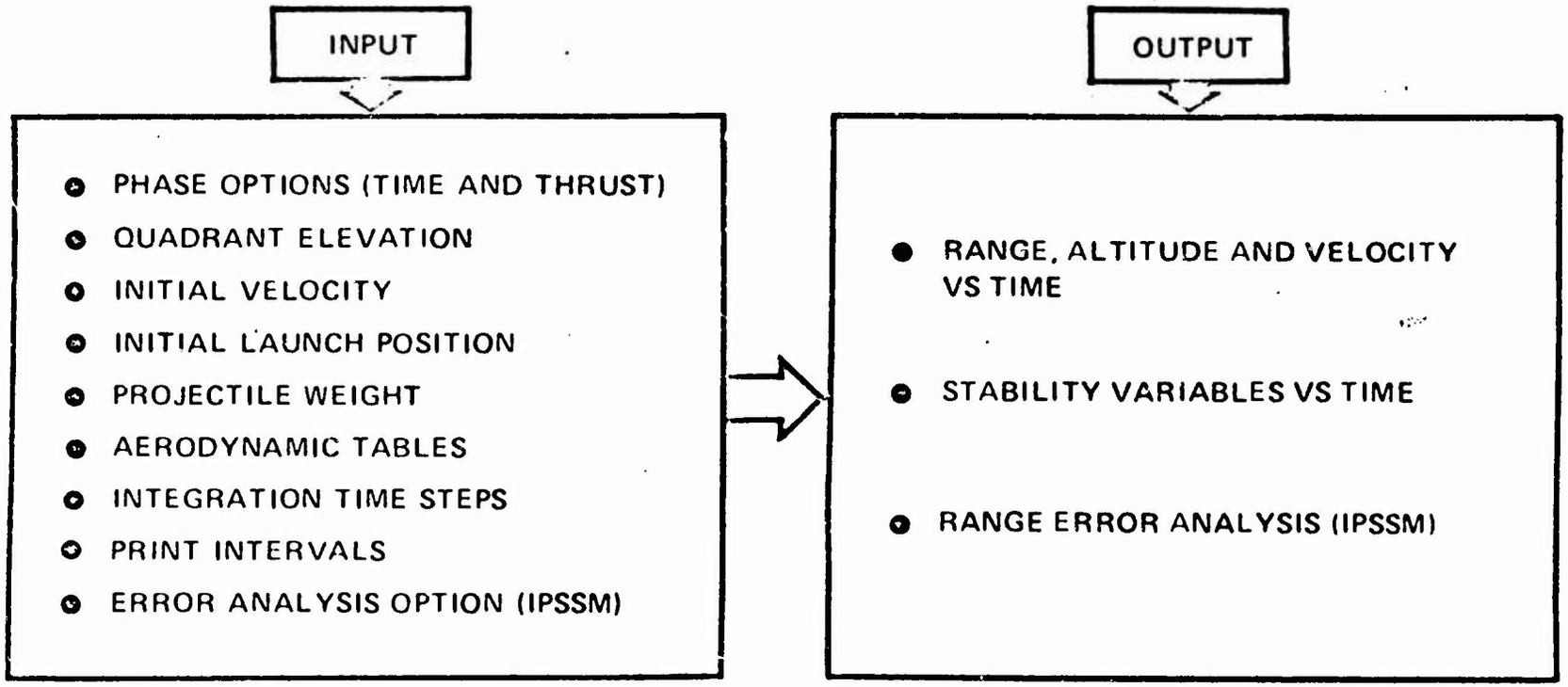
Vugraph 12

EXTERIOR BALLISTICS PROGRAM

- TWO STAGE, POINT MASS TRAJECTORY PROGRAM
- STABILITY AND DRAG CANCELLING OPTIONS
- UTILIZES STANDARD 1959 ARDC ATMOSPHERE OR NON-STANDARD CONDITIONS MAY BE ENTERED
- CAN INVOKE AUTOMATIC ERROR ANALYSIS (IPSSM)
- CONSTANT OR VARIABLE THRUST AND BURNING RATE

Vugraph 13

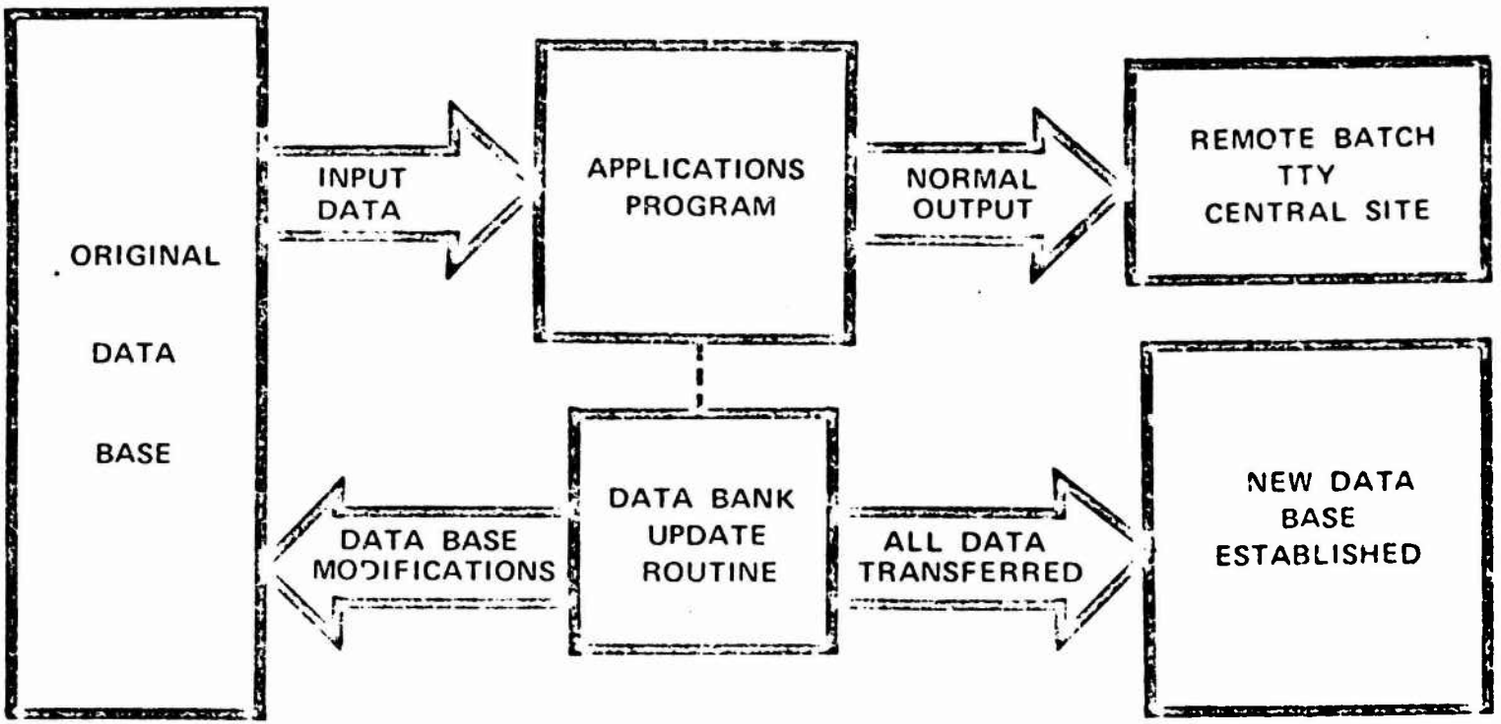
EXTERIOR BALLISTICS (INPUT-OUTPUT)



Vugraph 11

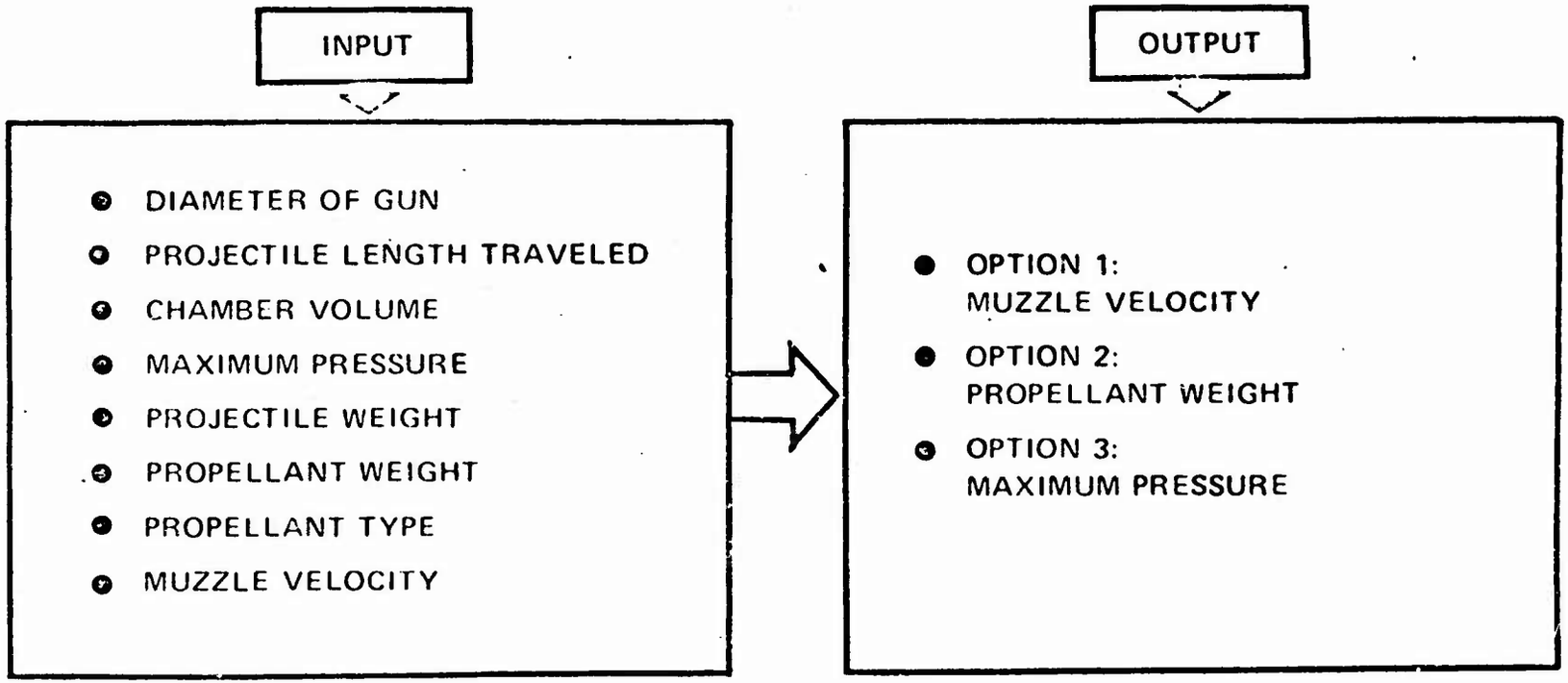
IPSSM

DATA TRANSFER SYSTEM



Vugraph 15

INTERIOR BALLISTICS
(SHORT VERSION)



TO INVOKE OUTPUT OPTIONS 1, 2 OR 3 SET CORRESPONDING INPUT VARIABLE EQUAL TO ZERO

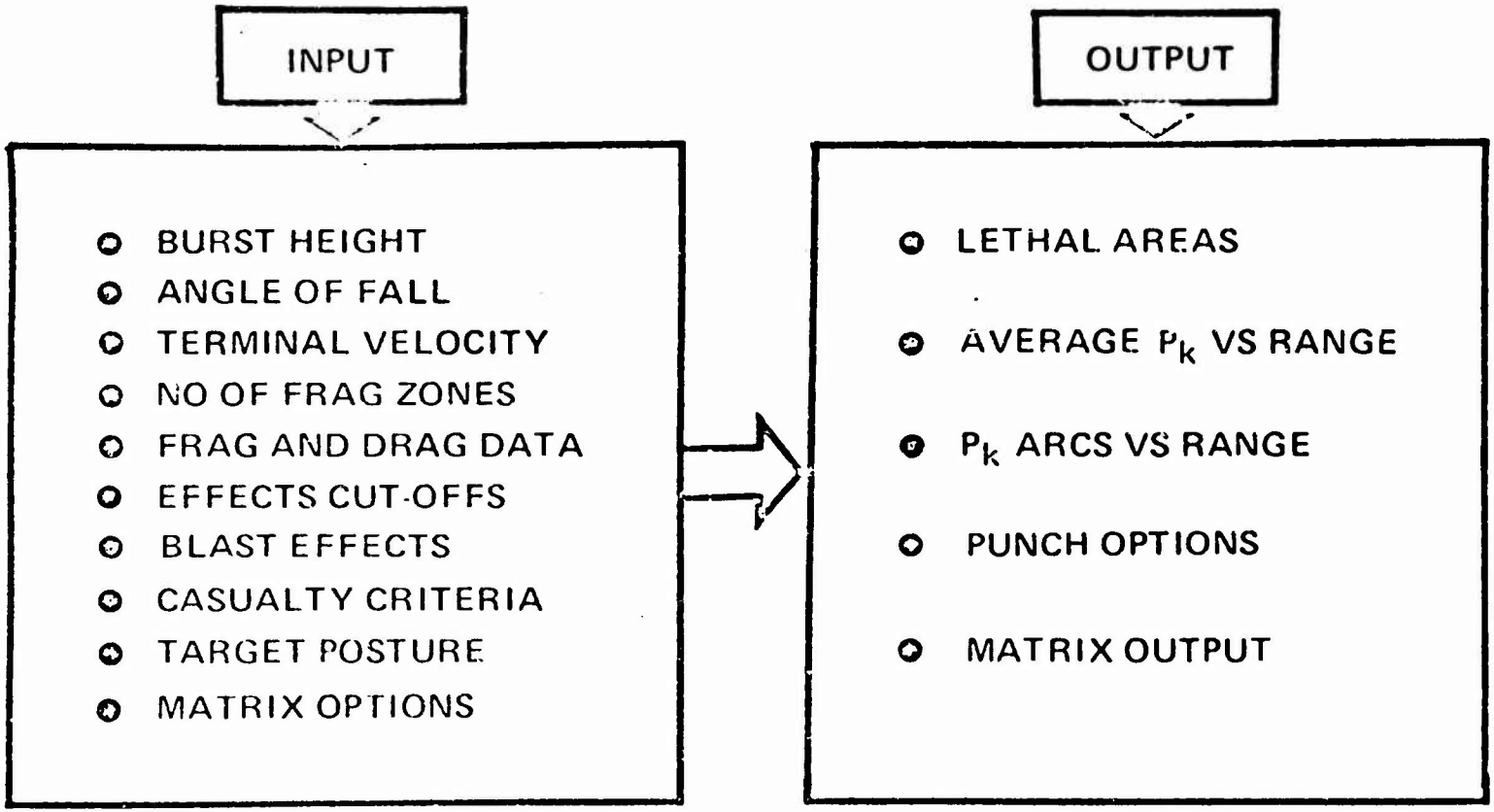
Vugraph 16

LETHAL AREA PROGRAM

- COMPUTES LETHAL AREA OF FRAGMENTATION AMMUNITION
- USES FRAG DATA IN TERMS OF CONICAL "ZONES" WITH RESPECT TO SHELL AXIS
- TAKES INTO ACCOUNT DRAG DATA THRU DIFFERENT MEDIA (AIR, GRASS, CANOPY, ETC)
- PERSONNEL TARGETS CAN ASSUME DIFFERENT POSITIONS (PRONE, STANDING OR IN FOXHOLES)
- MILITARY STRESS CONDITIONS MAY BE VARIED (DEFENSE, OFFENCE ETC)

Vugraph 17

LETHAL AREA (INPUT-OUTPUT)

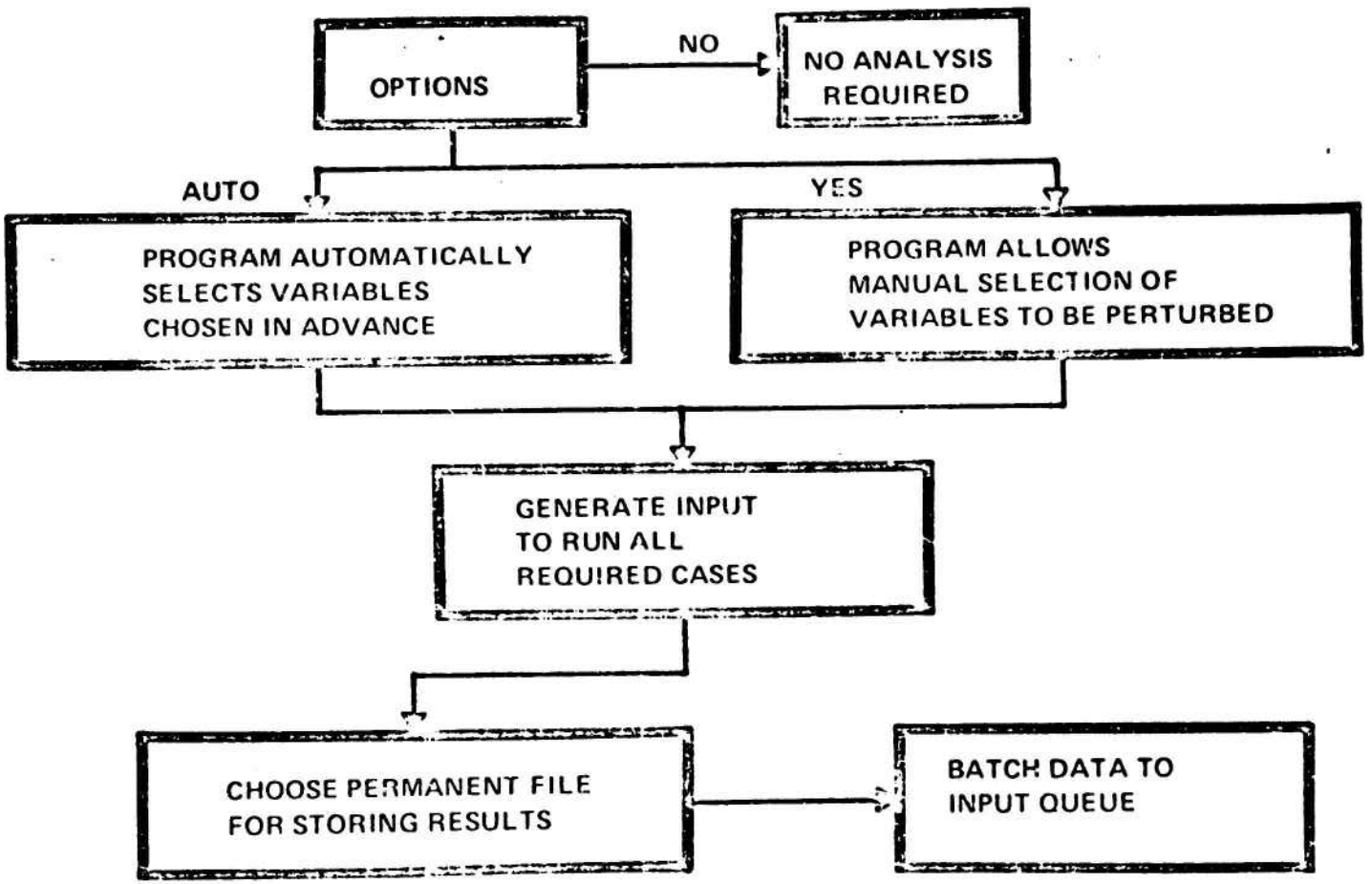


Vugraph 18

ERROR ANALYSIS SCHEME

(IPSSM)

EXTERIOR BALLISTICS (AR)



Vugraph 19

IPSSM

FY75 PLANS

- PROVIDE CAPABILITY TO CONDUCT EXTENSIVE OPTIMIZATION AND DATA ANALYSES
- IMPLEMENT COST EFFECTIVENESS MODULE
- EXTEND MODEL TO HANDLE A FULL COMPLEMENT PROJECTILE TYPES
- DEVELOP AND INCORPORATE SAFING, ARMING AND FUZING MODULES
- PROVIDE INTERACTIVE GRAPHICS MODE TO MAXIMUM EXTENT POSSIBLE
- MAINTAIN CURRENT SYSTEM TO INCLUDE TRAINING AND FULL DOCUMENTATION

Vugraph 20

BENEFITS OF IPSSM PROGRAM

- REDUCES NUMBER OF PROTOTYPES REQUIRED FOR TESTING
- REDUCES DECISION TIME
- INCREASES QUALITY OF WORK
- AVOIDS DUPLICATION
- PROVIDES CENTRAL SOURCE OF COMPUTER PROGRAMS
- IDENTIFIES TECHNOLOGICAL GAPS
- PROVIDES ADDITIONAL ENGINEERING TIME FOR INNOVATIONS
- REDUCES CHANCE OF PROTOTYPE FAILURE AND RETROFIT
- PROVIDES FOR BETTER COMPUTER PROGRAM DOCUMENTATION

Vugraph 21

NON-LINEAR AND MIXED INTEGER PROGRAMMING

Byron O. White
Management Information Systems Directorate
Mathematical Analysis Division
Picatinny Arsenal
Dover, New Jersey

ABSTRACT. The presentation will describe the experience which Picatinny Arsenal has in the field of linear programming and mixed integer programming. The talk will cover the nature of LP and MIP, application areas, and the use of computer codes at this Installation. Typical input/output and the description and handling of a rather large problem will be discussed.

INTRODUCTION. Linear programming encompasses a wide spectrum of users from the mathematically rigorous to those that are management oriented. The present day mathematical program system, as it is referred to, is oriented to both types of users as amazing as this may seem. These systems contain a base of involved matrix routines surrounded by efficient matrix and report generators. The efficiency obtained by these central memory and time consuming monsters requires that every advantage obtainable from a given computer system be utilized.

Recent developments allow the linear programming user to restrict some of the variables of a problem to take on only integer values - hence the name mixed integer programming. Although the number of integer variables allowed is small, the increased capability to the user is extraordinary.

This presentation will neither elaborate on the mathematical aspects nor on the results, but rather deal with some of the applications and basic concepts generally used in solving problems.

Mixed integer programming may be used in many different application areas ranging from executive decision in the front office to efficient methods of packaging and handling in the shipping department. The engineer and scientist may also use mathematical programming systems to solve technical problems from control of nuclear reactors to solution of large matrices generated by PDE and other related problems. What then are some specific areas of application?

Production scheduling problems dealing with multiple product assembly lines, changeover costs, and time phased assembly.

USES OF MIXED INTEGER PROGRAMMING

- PRODUCTION SCHEDULING
- INVESTMENT TRADE-OFFS
- INVENTORY CONTROL
- SITE SELECTION
- TRANSPORTATION AND DISTRIBUTION
- CHEMICAL BLENDING

Figure 1

Investment tradeoff where the decision maker must choose between risk and payoff, or for stock market enthusiasts, the choice between portfolio buying or individual stock transaction and manipulation.

Inventory control where the problem is a combination of knapsack problem and investment problem.

Site selection deals with choices of raw material acquisition as well as building cost indices. This type of problem is related to a transportation problem.

Transportation and distribution considers various modes of transportation, time relationships, and physical properties of products.

Chemical blending such as in the operation of a refinery where there is a requirement and associated cost for every product.

In the case of most Army problems as is the case of most industrial problems, it takes a combination of these methods to successfully solve most problems.

Not all problems due to their structure are solved the same way. Hence there is no one best algorithm to solve all problems. What then are some of these solution methods (Figure 2)? In the case of small problems, in other words where the number of integer variables are small, the methods of solution used are Cutting Plane and Enumeration. Enumeration solves the problem for all possible values for each of the integer variables and then manually or by some snazzy algorithm ranks the various alternatives. The Cutting Plane methods, of which Gomorey's algorithm is one, are geometric in form and usually difficult to control, although in some cases one can find rapid solutions to small problems.

Bender's algorithm is a method used on medium-sized problem, medium-sized meaning problems where the Cutting Plane method runs into difficulties in obtaining a solution. In Bender's algorithm the problem is separated into two parts, an integer part and a continuous part. These two parts are solved separately and then solutions are obtained by putting the two parts back together. If this all sounds complicated, it is.

THEORY OF MIXED INTEGER PROGRAMMING

SMALL PROBLEMS

- ENUMERATION
  - CONSIDERS EVERY POSSIBLE INTEGER SOLUTION
- GOMORY CUTTING PLANE METHOD

MEDIUM - SIZE PROBLEMS

BENDER'S ALGORITHM

- INTEGER AND CONTINUOUS PARTS OF PROBLEMS ARE SOLVED SEPARATELY

LARGE PROBLEMS

BRANCH AND BOUND

- BRANCH - POSSIBLE VALUE FOR AN INTEGER VARIABLE
- LP SOLUTION FOR EACH BRANCH CHOICE
- CRITERIA FOR BRANCH ELIMINATION

Figure 2

TYPES OF INTEGER VARIABLES

BIVALENT OR DECISION — 0 - 1

INTEGER OR QUANTITY — RANGE

SPECIAL ORDERED SETS

TYPE 1 - ONLY ONE VARIABLE IN A SET MAY BE NON-ZERO

TYPE 2 - ONLY TWO ADJACENT VARIABLES IN A SET MAY BE NON-ZERO

Figure 3

Most large scale mixed integer mathematical programming codes use the Branch and Bound algorithm. This is so because it has been found to be the most stable code for the solution of large problems. Each manufacturer adds his own specialties which again make the code very problem dependent but basically they are all the same.

The Branch and Bound algorithm naturally lends itself to some very special variable types (Figure 3). The first type is the bivalent or decision variables. It can take on the integer values of 0 or 1. This variable allows the user to incur fixed costs or make decisions during solution of the problem. The second type is the integer or quantity variable. This variable indicates that in the solution only an integral quantity is allowed. Most codes require a range in which this integral value must lie which can effect running time. The last of these three types are known as special ordered sets. Special ordered sets are fancy ways of using bivalent variables. There are two types of special ordered sets. Type one is where only one member of the set may be set to one and all the rest set to zero. Type two is where two adjacent members must be set to one and all the rest set to zero. These sets may be used to handle non-linear relationships as well as discontinuous types of constraints.

An example of a simple text-book type problem is shown in Figure 4. This problem has a non-linear constraint where the feasible region is bounded by the elliptical constraint and various values of the objective function are represented by the parabolic curves. In separable programming, care must be taken that all functions be kept convex so that a solution may be obtained. However, by using type two special ordered sets, this need not be the case. In order to solve a problem such as this, the user rewrites the equations in a linearized form represented in Figure 5 and solves the resulting LP problem. In this formulation the top equation represents the constraint and the bottom equation represents the objective function. The λ represents points on a linearized curve (Figure 6). Formulation of this problem using bivalent variables or special ordered sets results in the solution shown in Figure 4.

Next I would like to discuss a problem presently being solved. This problem is a combination of transportation, production, site selection, and investment problems encountered in modernization of the existing army production base (Figure 7). An integrated line is an assembly line which produces all the components for a given ammunition item, and a complex is defined to be a plant containing multiple item integrated

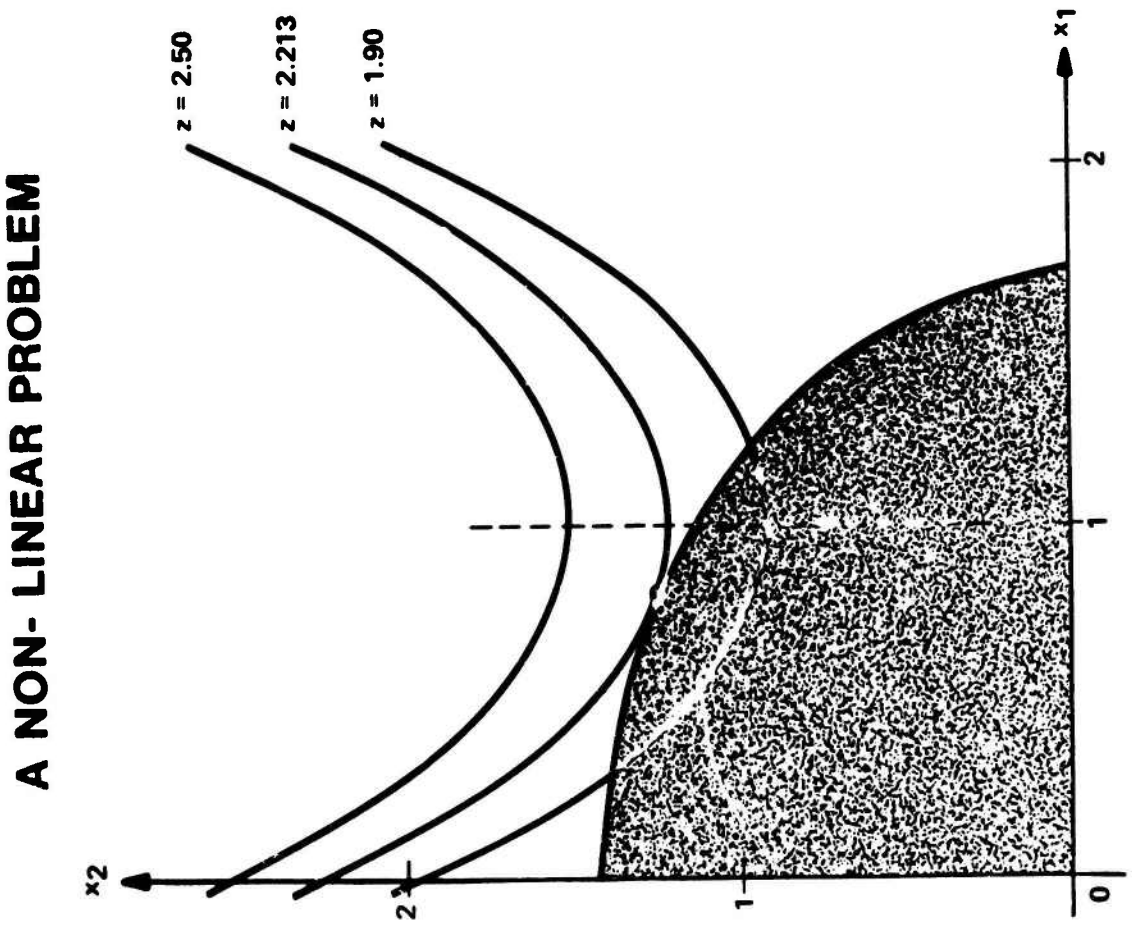


Figure 4

LINEARIZED VERSION OF PROBLEM CONSTRAINTS

$$\sum_{j=1}^{2}\sum_{k=0}^{8} g_{kj}\lambda_{kj} + x_3 = 6,$$

$$\sum_{k=0}^{8}\lambda_{k1} = 1,$$

$$\sum_{k=0}^{8}\lambda_{k2} = 1,$$

$$\lambda_{k1} \geq 0, \quad \lambda_{k2} \geq 0, \quad \text{all } k, \quad x_3 \geq 0,$$

$$\max z = \sum_{j=1}^{2}\sum_{k=0}^{8} f_{kj}\lambda_{kj},$$

Figure 5

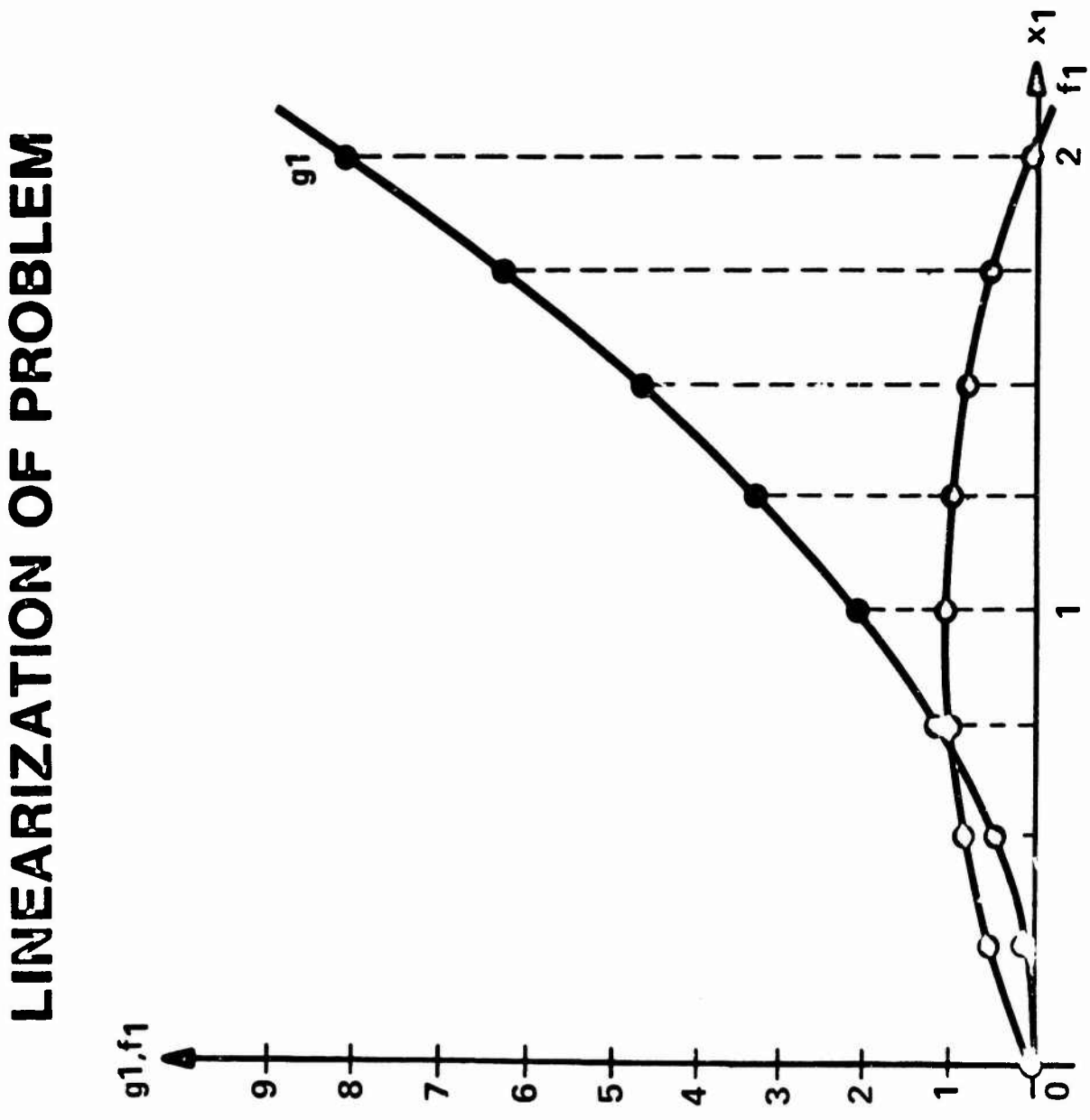


Figure 6

lines. It is important to note that the existing ammunition production base is structured such that a solution for only a single ammunition item is not sufficient for production planning purposes. Therefore the base restructure economic model solves the problem for multiple end items to determine the most economic mix of plant production.

In Figure 8 it can be seen that a trade-off exists between a complex or integral plant versus an existing plant where component parts must be shipped in. Also a trade-off exists between in-plant storage versus shipping through a depot. In this problem both a Pacific and European requirement were used. All of the trade-offs are based on economic or cost considerations as shown in Figure 9. An example of the 105mm M1 production base along with a typical choice of plant and transportation configuration is shown in Figure 10.

It would appear that although very large LP problems may be solved using mixed integer codes, the number of allowable integers is extremely limited. By careful formulation and choice, a few integer variables can be made to control many other variables. Figure 11 shows some specific details of how this was accomplished in the case of the base restructure economic model.

The top of the chart represents some typical constraints and the bottom represents some typical results. I_t is a bivalent variable representing when a plant is operating in time period t. B_t is a bounded continuous variable representing a time period in which a plant is built or modernized. O_t is a bounded continuous variable representing when a plant is opened. K_t is a bounded continuous variable representing when a plant is closed. From the possible results it can be seen how the integer variables I_t force the variables B_t O_t , K_t to also be integers. Constraint 2 says once B_t is set to 1 it must remain 1. This allows (C_t-C_t+1) of the cost function to represent a cascaded building cost which in this case represents the discounted cost of building the plant in different time periods. C_L in the cost function represents the cost of the plant while in lay away and since it occurs in both the B_t and I_t variables, this cost is only incurred after the plant has been built but only when its not producing. C_O represents a cost of opening a plant or taking it out of lay away where C_K represents the cost of putting a plant into lay away. C_I represents the fixed cost of operating a plant which in this case in strictly the overhead cost. Other factors such as minimum and maximum operating rates may also be controlled by the same I_t variable.

BASE RESTRUCTURE
ECONOMIC MODEL

- USES INVESTMENT COSTS, OPERATING COSTS, AND TRANSPORTATION COSTS TO DETERMINE THE MOST ECONOMIC AMMUNITION PRODUCTION BASE. THE MODEL PERFORMS MULTIPLE ITEM SYSTEM ANALYSIS CONSIDERING AMMUNITION COMPONENT PRODUCERS, LAP PLANTS, INTEGRATED LINES, COMPLEXES, DEPOTS, PORTS, AND OVERSEAS THEATER REQUIREMENTS

- THE MODEL OUTPUT DISPLAYS THE OPTIMUM PLANT CONFIGURATION(S) PLANT LOCATION(S), AND THE COST OF OPERATION FOR PRODUCING EACH ITEM OF AMMUNITION

Figure 7

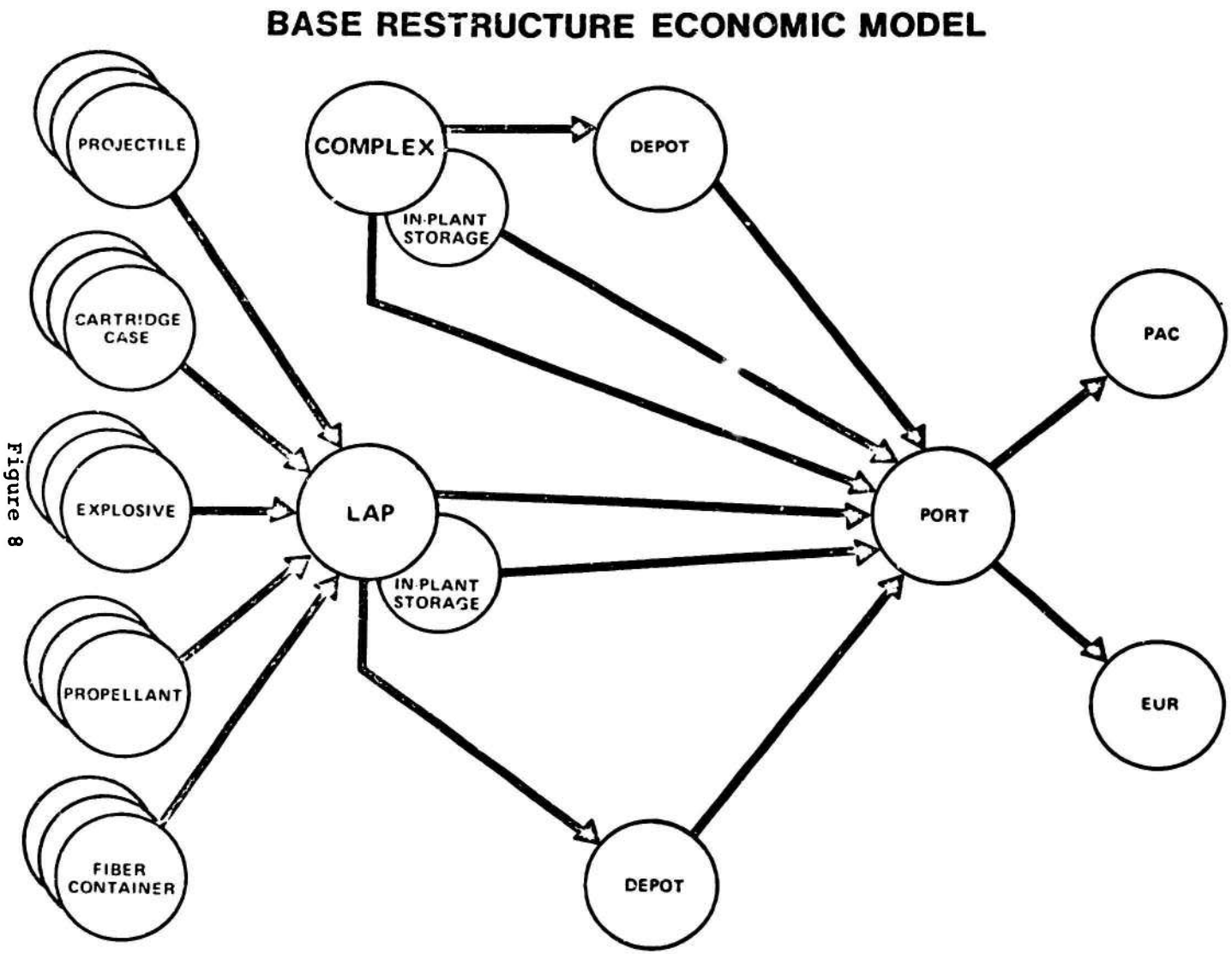

BASE RESTRUCTURE ECONOMIC MODEL
PROJECTILE
CARTRIDGE CASE
EXPLOSIVE
PROPELLANT
FIBER CONTAINER
COMPLEX
IN-PLANT STORAGE
LAP
IN-PLANT STORAGE
DEPOT
DEPOT
PORT
PAC
EUR

Figure 8

TRADEOFF ANALYSIS OBJECTIVE

TO DETERMINE THE OPTIMUM ECONOMIC MIX OF PLANTS, COMPLEXES, AND INTEGRATED LINES FOR AMMUNITION PRODUCTION TO MEET MOBILIZATION REQUIREMENTS BASED ON PEMA POLICY & GUIDANCE FOR MODERNIZATION AND EXPANSION. CONSIDERATION WILL ALSO BE GIVEN TO CAPABILITY TO ASSESS PEACETIME REQUIREMENTS.

Figure 9

BASE RESTRUCTURE
ECONOMIC MODEL - 105MM HE M1 CARTRIDGE

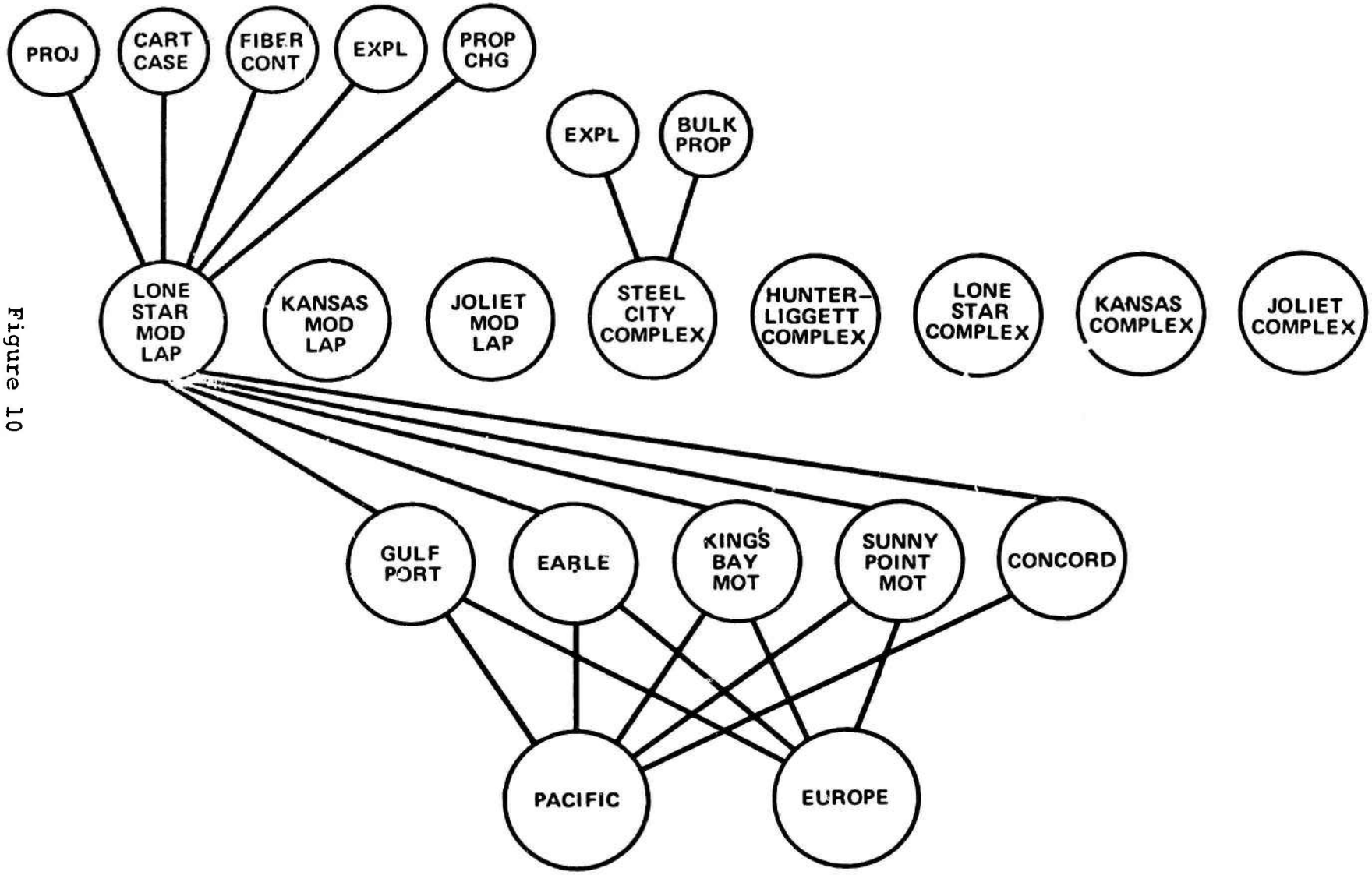


Figure 10

CONSTRAINTS USING INTEGERS

$$B_t \geq I_t$$

$$B_{t+1} \geq B_t$$

$$I_t - I_{t-1} \leq O_t$$

$$I_{t-1} - I_t \leq K_t$$

COST FUNCTION (MINIMIZE)

$$(C_t - C_{t+1} + C_L)\, B_t + C_o O_t + (C_i - C_L)\, I_t + C_k K_t$$

POSSIBLE RESULTS

	t_1	t_2	t_3	t_4	t_5
I_t	0	1	1	0	1
B_t	0	1	1	1	1
O_t	0	1	0	0	1
K_t	0	0	0	1	0

Figure 11

CONSIDERATIONS

SIZE AND FORMULATION OF PROBLEM

- SIZE AND DENSITY OF MATRIX
- AMOUNT OF DATA REQUIRED
- TYPES OF VARIABLES IN MODEL

COMPUTER RESOURCES

- COMPUTER HARDWARE AVAILABLE
- MATH PROGRAMMING CODE REQUIRED
- OPERATING SYSTEM AVAILABLE

SOLUTION TOLERANCES

- ACCURACY OF RESULTS (CEFGW)
- CODE TOLERANCES

Figure 12

Figure 12 represents parameters to be considered when formulating large LP problems. It should be remembered that modern LP codes really represent a small fast running algorithm surrounded by a huge data handling system.

TARGET LOCATION USING AN ARRAY OF SENSORS WHICH PRODUCE CLOSEST POINT OF APPROACH AND MULTIPLE RANGE ALARMS

Raymond F. Coakley, Jr.
Special Projects Division
Countermine/Counter Intrusion Department
Mobility Equipment Research and Development Center
U. S. Army Troop Support Command
Fort Belvoir, VA 22060

ABSTRACT. The problem of target location by means of an array of sensors can be approached in many ways. Most ways require the processing of analog data at some central point. The objective of this approach is to use two distinct outputs of omnidirectional point sensors generated in response to (1) a target coming within range, and (2) passing at CPA (Closest Point of Approach). Each sensor is designed to emit a coded alarm indicating the time of passage at several range (gain) thresholds and another coded alarm indicating time at CPA, through processing of acoustic, seismic or other signals. The sensor alarms are received and clocked at some central location (SRU) where a coded message is passed to a data processing computer. This method utilizes very low information bandwidths compared to the analog data, thus, it is suitable for use in a sophisticated electronic countermeasure environment.

The author has developed a mathematical model by which the necessary target parameters of velocity and position can be calculated. Location accuracy depends on sensor position error, CPA alarm time error, and range ratio error. For the worst combinations of errors, the resultant target parameters are found to be marginally acceptable. However, in the typical case where the errors are randomly distributed, the target velocity and location are usually specified with sufficient accuracy for Army target location requirements. The combined effect of all errors on the specification of target position one minute after the target passes through the array produces an error of less than 75 meters in the average (randomly simulated) case.

The use of a sensor producing a pair of range alarms such that the ratio of the two ranges is constant has not been previously reported as a method of avoiding the problem of unpredictable range. The variations caused by energy propagation, various source intensities, and different transducer implacements are greatly reduced as the need for knowing exact target ranges is eliminated by this range ratio technique.

1. INTRODUCTION. The subject matter of this paper emerged from a program to develop ground sensors capable of producing information useful for locating and classifying military vehicles on and above the battlefield. The Special Projects Division is now building sensors which can locate wheeled and tracked vehicles when used in certain arrays and processing schemes.

2. APPROACHES. An array of unattended ground sensors is capable of providing target location information if their locations are known and if at least four non-redundant target determined alarms are produced. The need for four independent data values can be seen in the expression for the target coordinates as a function of time. (Figure 1.) The four alarms which provide sufficient data are identified with some distance relationship between the target and the sensor. Their times of arrival at a central location are the data values. Sensors producing alarms corresponding to three distinct target-to-sensor relationships are the object of this discussion.

The most basic of these sensors activates only at a specified range to produce one kind of alarm. (Figure 2.) Four of these one-range sensor alarms produce data which can be used to solve four simultaneous second-order equations for the target parameters of velocity and time-dependent coordinates. The range of each of the sensors as well as their coordinates must be known. A computer solution was written for the set of equations which showed a very great sensitivity to incorrect range values. With present technology the unreliability of the range specification adds to the complexity of the equations and makes this approach impractical.

The second of these sensors, a more sophisticated one, activates with a unique alarm corresponding to the time when a target is at its closest point of approach (CPA) to the sensor. (Figure 3.) Three of these CPA sensor alarms provide sufficient data to calculate target velocity. (Figure 4.) Any pair of velocity equations can be solved for speed and bearing if the target motion is assumed linear. If one of these can also produce a fixed range alarm, then the complete target location can be calculated. (Figure 5.) The arctangent of the bearing can be found, then the speed;and this speed value leads to the sensor-to-CPA distance which locates the target path. The equations are simpler and the minimum number of sensors is smaller; but the previously mentioned unreliability of the range specification makes this approach defective in practice, too.

Another type of sensor is being developed to reduce the range determination problem. It appears that a sensor can be built which will produce a pair of range alarms such that the ratio of the two ranges is reasonably constant even though the actual ranges vary greatly. (Figure 6.) The range ratio replaces the range, and the coordinates of the CPA do not depend explicitly on this unreliable parameter. (Figure 7.) Thus the variations caused by energy propagation through the medium, by the various source intensities of different targets, by different transducer emplacement, etc. can be reduced. By increasing the minimum data points to five, the range parameter is not specifically needed. Such a multiple range sensor, now called a CPA/MR sensor, is an object of intense development.

3. DISCUSSION OF METHOD. The solution program for an array of eight or more CPA/MR sensors in a field array has been written and debugged. The target coordinates are expressed as a function of time and a straightforward application of the Pythagorean Theorem and the trigonometric relations yields enough equations to solve for the two components of velocity and some reference location from the input sensor locations, range ratios and activation times. (Figure 8.) Because a real system can not be expected to produce all the appropriate alarms from all sensors, the program was written to take the alarm times available from an arbitrary number of sensors and use them in all possible combinations to compute average speed, average bearing and least-squares-fitting target path.

The average target location is found as follows. Each of the CPAs is used to find the target coordinates at an arbitrary "reference" time. Since a CPA can be found for each sensor which activates with the full complement of Range 1, Range 2 and CPA alarms, a number of paths can be found. The target coordinates are calculated for the last CPA time, and the mean values of these coordinates is the equivalent of a least-squares fit of the target path.

Several features of the target location program are worth noting. While the data and geometry provide a unique solution for the velocity from the sensor coordinates and CPA alarm times, the range ratio times provide an ambiguous CPA. The application of the Pythagorean Theorem in the calculation of the distance from sensor to CPA produces a square root value. The ambiguous polarity corresponds to the possibility of the target's path being located on either side of the sensor. (Figure 9.) The resolution of this ambiguity constituted a major problem in the algorithm development. The method found to perform most consistently calculates the distances (D and D') from the two possible CPAs [(X,Y) and (X',Y')] to the other sensor locations [e.g., (X_j,Y_j)], one after the other; then it determines which distance comes closest to the ideal hypoteneuse of the triangle formed by the other sensor's offset (A_j) and the target path length [$V(t_{ic} - t_{jc})$].

In the operational solution program some other finesse is introduced. The target location which arises from the average values coming from various combinations of redundant data reflects certain arbitrary parameters. These are used to determine the acceptibility of intermediate values in the mathematical solution, for example, a minimum time difference or an individual bearing value's contribution to an average. These allow adjustment of the program to various levels of sensitivity to input variable deviations, as well as to reasonable speeds of the expected targets. In addition, the counting parameters used in the averaging processes are retained as a measure of confidence in the input data and of "goodness of fit" of the target path.

4. TESTING BY SIMULATION. However, the target location program is only the first step in analyzing the performance of a sensor array in the real world. It must be tested against all reasonable target speeds and directions and in all expected configurations and dimensions. Then it must be tested for sensitivity to systematic and random errors in the input variables. This testing involved the largest portion of computer time and analyst energy.

A data generating program was written to compute the activation times that would result from a constant velocity target passing through an array at a chosen offset from some arbitrary point. Then the amount of variation in the sensor coordinates, range ratios and activation times could also be chosen. (Figure 10.) This error analysis was done in two stages, the first to determine what the worst case result could be and the second to determine what the average performance would be when the input variables were subject to random errors of specified deviations.

To find the configuration of input deviations which caused the worst output errors, an iterative technique rather than a mathematical analysis was used, since the solution equations were quite involved. This involved varying each of the input variables in turn, by a fixed amount, in all possible ways. Once the worst case was found, the deviations were increased systematically to plot the resultant errors in target velocity and coordinates.

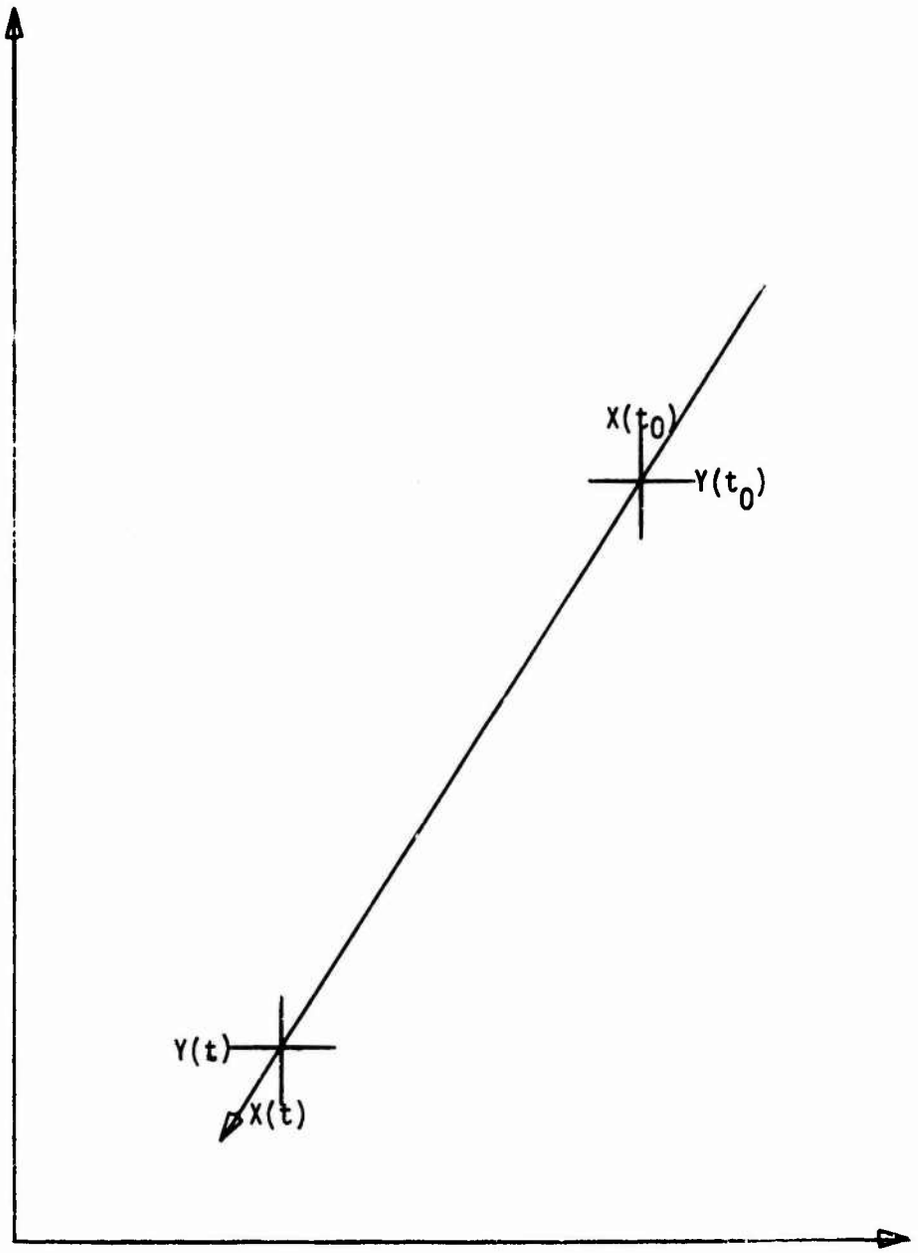
To find the average performance of the array, the input variables were calculated for a specified target approach, then were modified by a random error using a Monte Carlo technique. The averages were found to "settle down" after one or two hundred target passes, depending upon the size of the sigma assigned to the deviations.

The data generating program also made possible a trade-off determination of the best choice for range ratio and sensor separation. For example, greater sensor separation improves velocity accuracy but degrades path location. In addition, the best values for the arbitrary constants governing sensitivity could be found from the simulation program.

5. CONCLUSION. The net result of the development of the target location algorithm and the simulation program has been the determination of the accuracy which can be expected of an array of UGS as a function of sensor performance and the establishment of a working solution which can be incorporated in a remotely monitored battlefield sensor system.

$$X(t)=X(t_0)+V_x(t_0-t)$$

$$Y(t)=Y(t_0)+v_y(t_0-t)$$



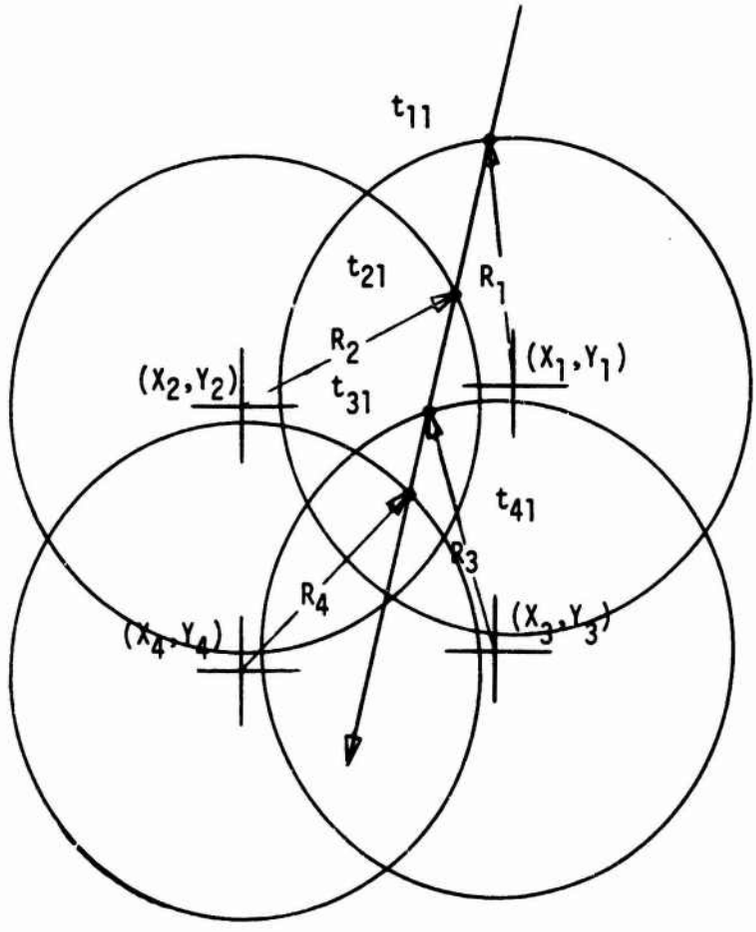
TARGET PATH GEOMETRY

FIGURE 1

$$R_i^2=(X_i-X(t_{i1}))^2 + (Y_i-y(t_{i1}))^2$$

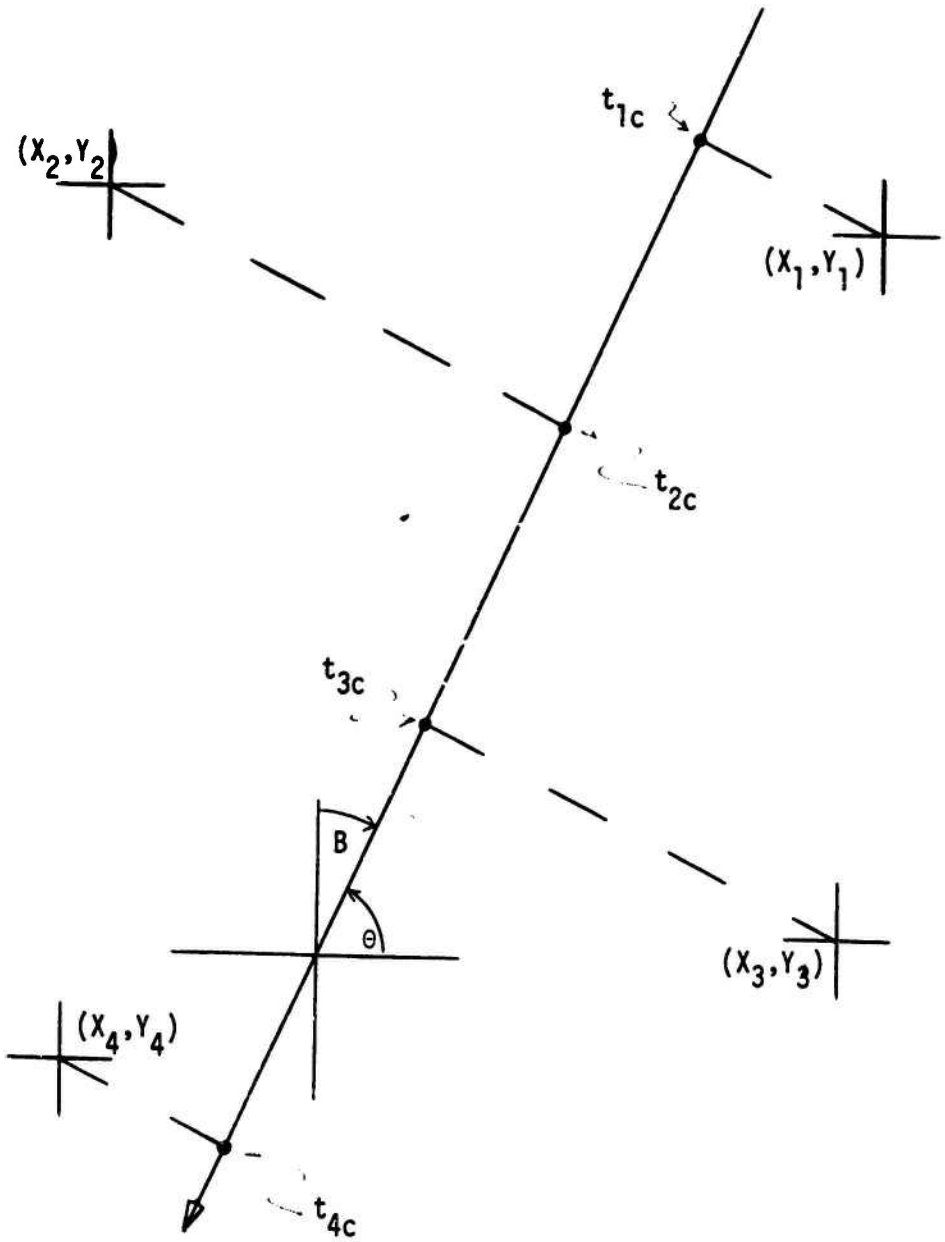
$$R_i^2=(X_i-X(t_0)-V_x(t_0-t))^2 + (Y_i-Y(t_0)-V_y(t_0-t))^2$$

$$R_i^2=(X_i-X(t_0)-V_x(t_0-t_{i1}))^2+ (Y_i-Y(t_0)-V_y(t_0-t_{i1}))^2$$



TARGET PASSING RANGE ALARM SENSORS

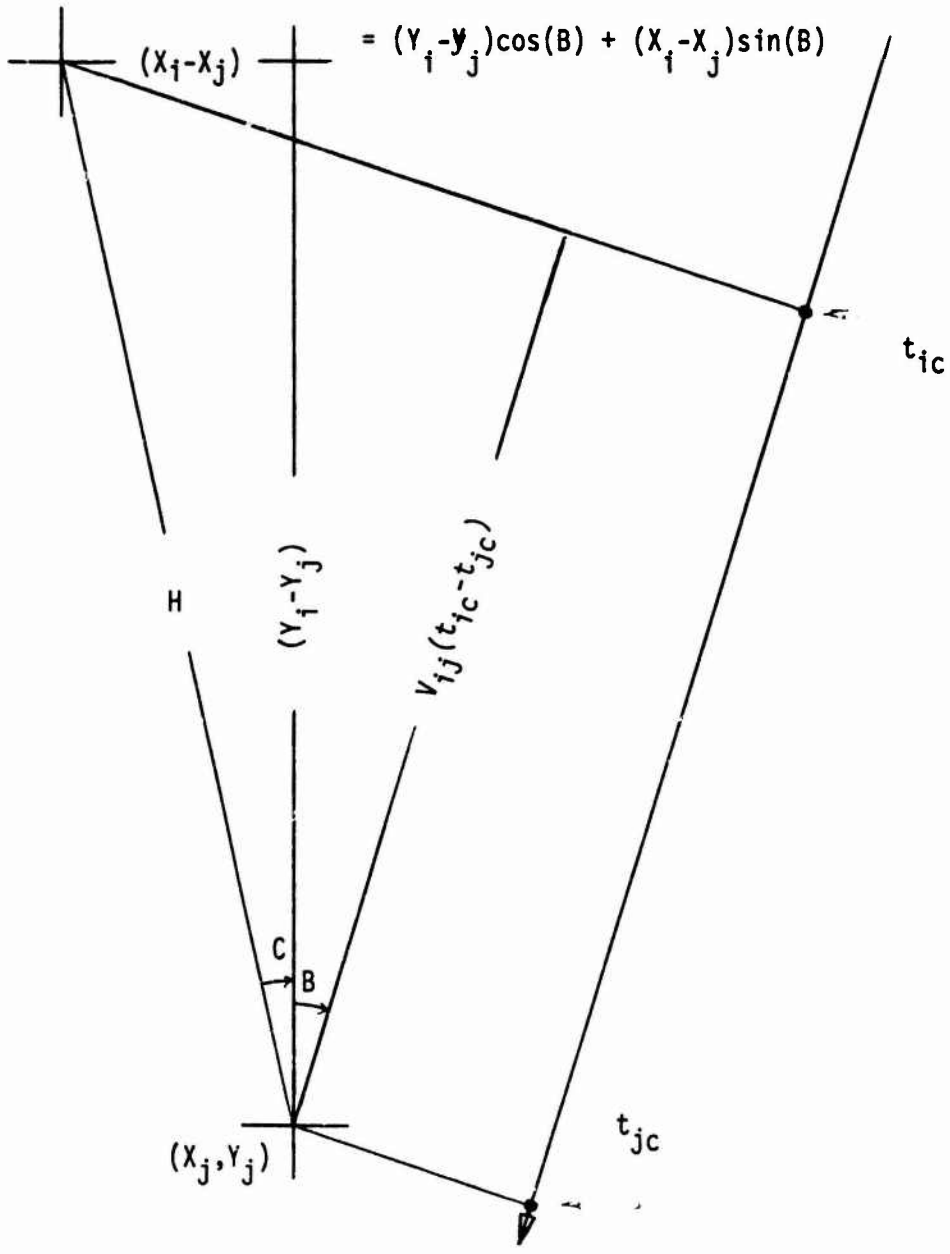
FIGURE 2



TARGET PASSING CPA ALARM SENSORS

FIGURE 3

$$V_{ij}(t_{ic}-t_{jc}) = H\cos(B-C)$$
$$= H(\cos(B)\cos(C) + \sin(B)\sin(C))$$
$$= (Y_i-Y_j)\cos(B) + (X_i-X_j)\sin(B)$$

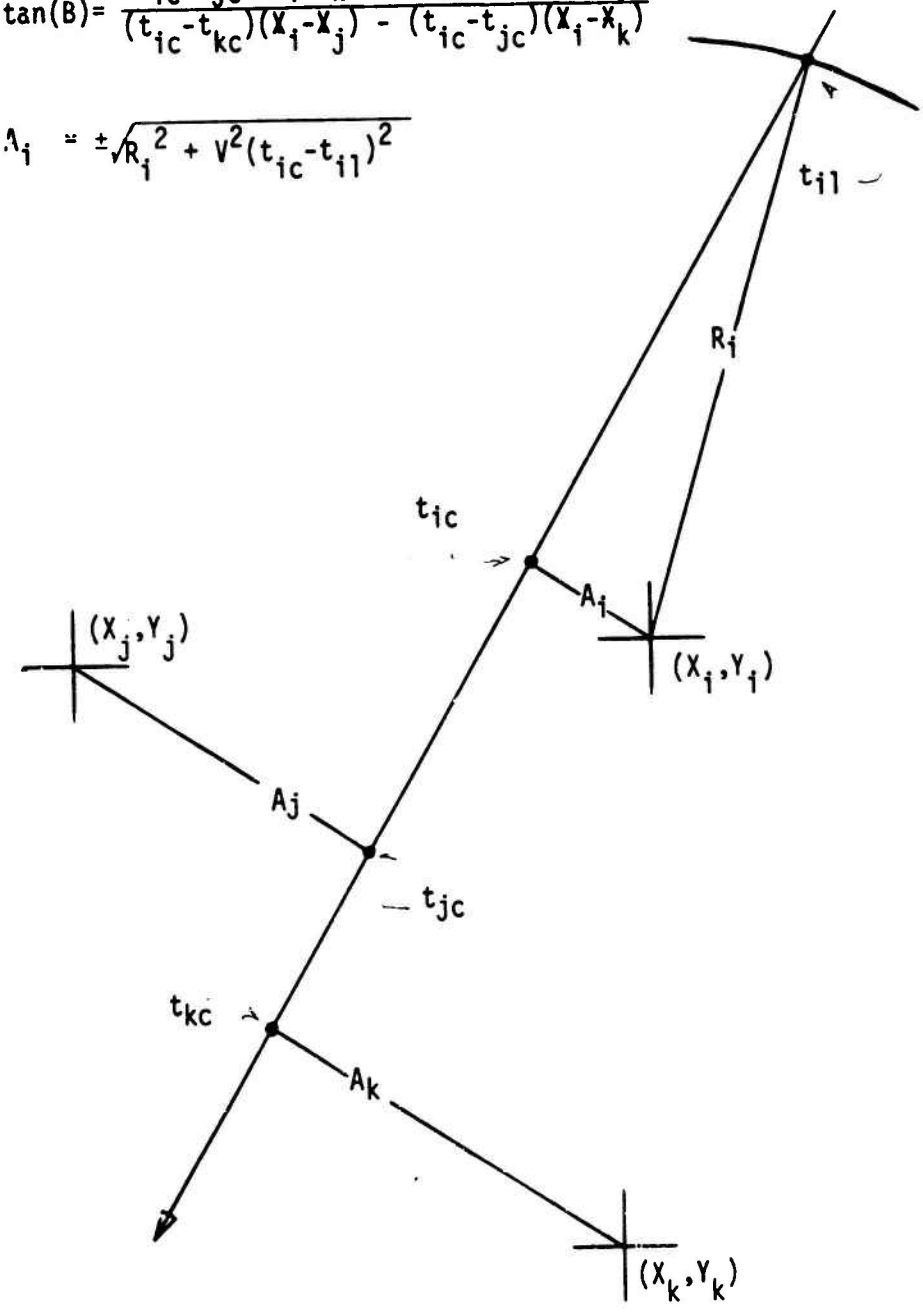


TARGET PATH TO VELOCITY RELATIONSHIP

FIGURE 4

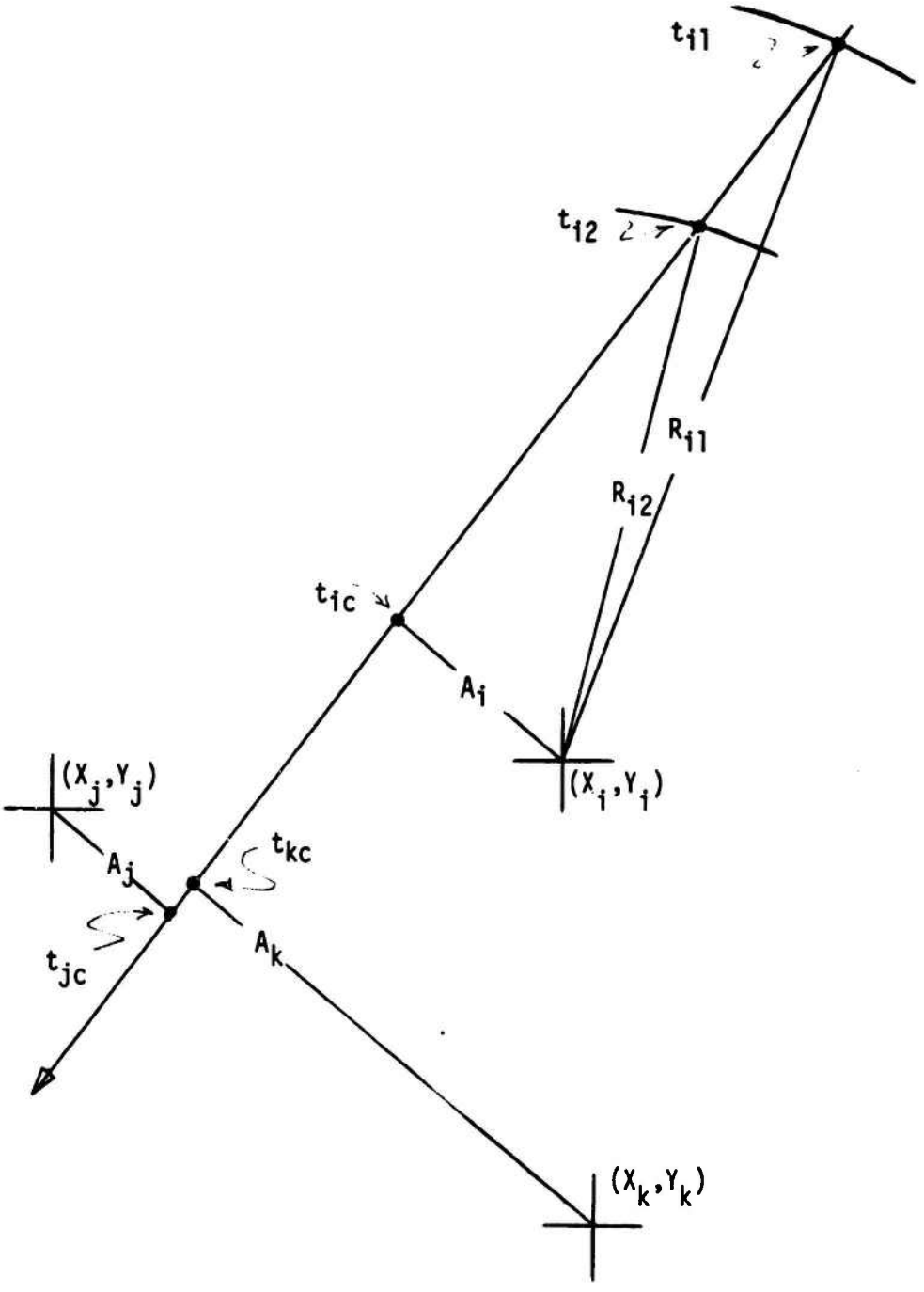
$$\tan(B) = \frac{(t_{ic}-t_{jc})(Y_i-Y_k) - (t_{ic}-t_{kc})(Y_i-Y_j)}{(t_{ic}-t_{kc})(X_i-X_j) - (t_{ic}-t_{jc})(X_i-X_k)}$$

$$A_i = \pm\sqrt{R_i^2 + V^2(t_{ic}-t_{il})^2}$$



TARGET PASSING CPA/RANGE ALARM SENSORS

FIGURE 5



, TARGET PASSING CPA/MULTI-RANGE ALARM SENSORS

FIGURE 6

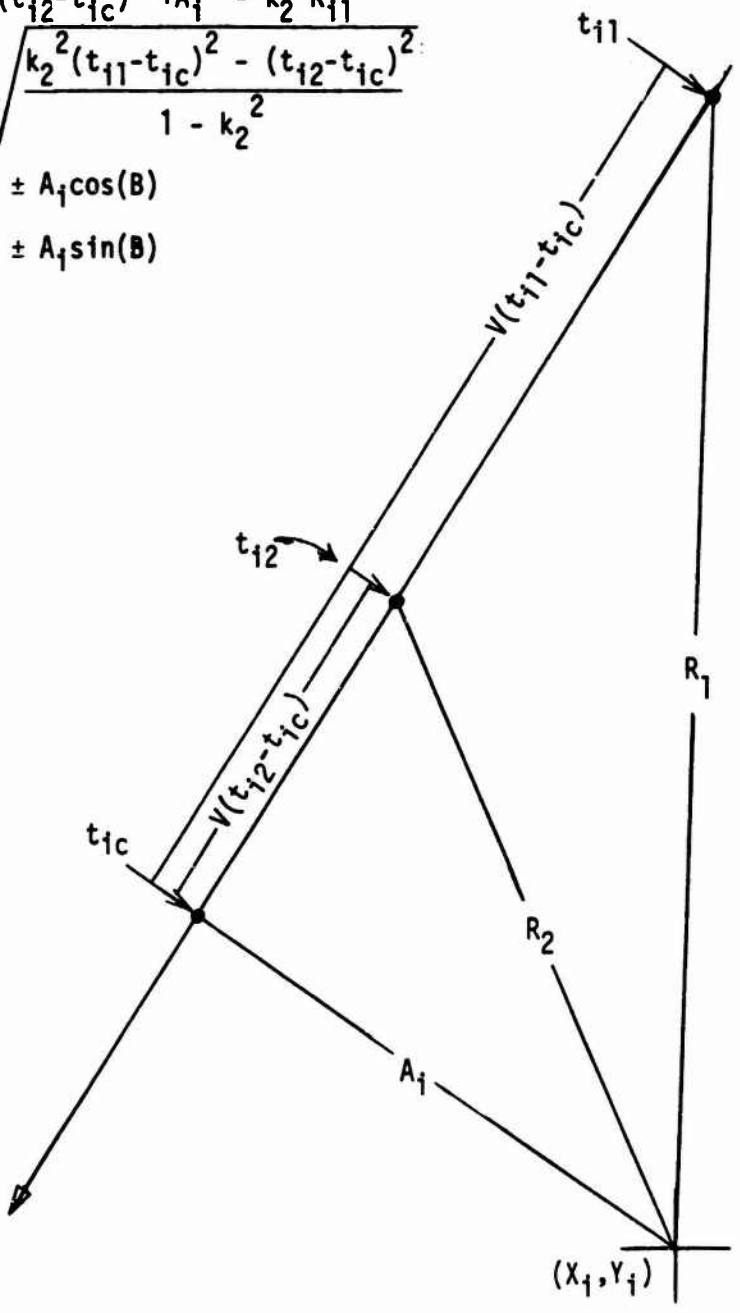
$$R_{i1}^2 = V^2(t_{i1}-t_{ic})^2 + A_i^2$$

$$R_{i2}^2 = V^2(t_{i2}-t_{ic})^2 + A_i^2 = k_2^2 R_{i1}^2$$

$$A_i = \pm\Psi\sqrt{\frac{k_2^2(t_{i1}-t_{ic})^2 - (t_{i2}-t_{ic})^2}{1 - k_2^2}}$$

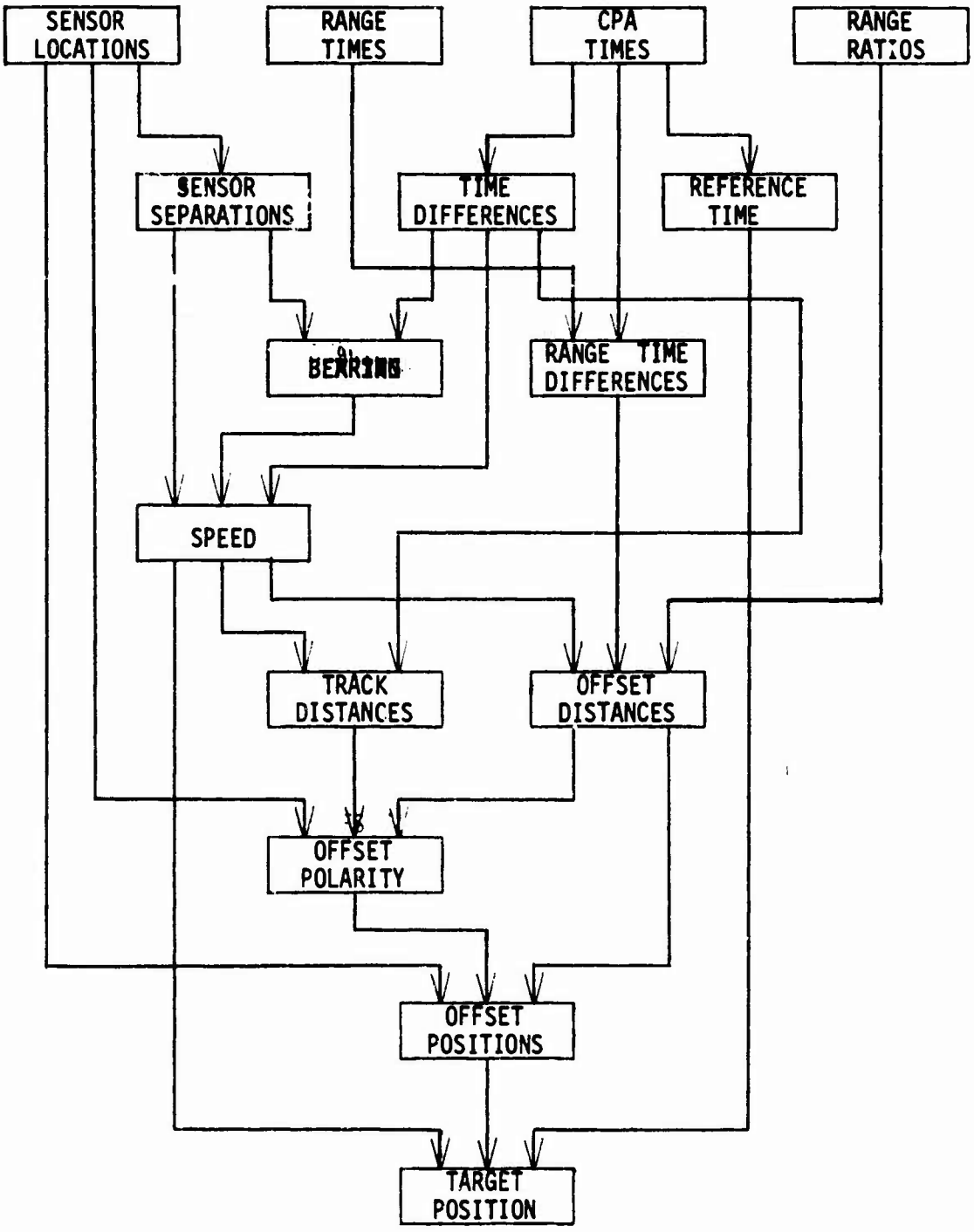
$$X(t_{ic}) = X_i \pm A_i\cos(B)$$

$$Y(t_{ic}) = Y_i \pm A_i\sin(B)$$



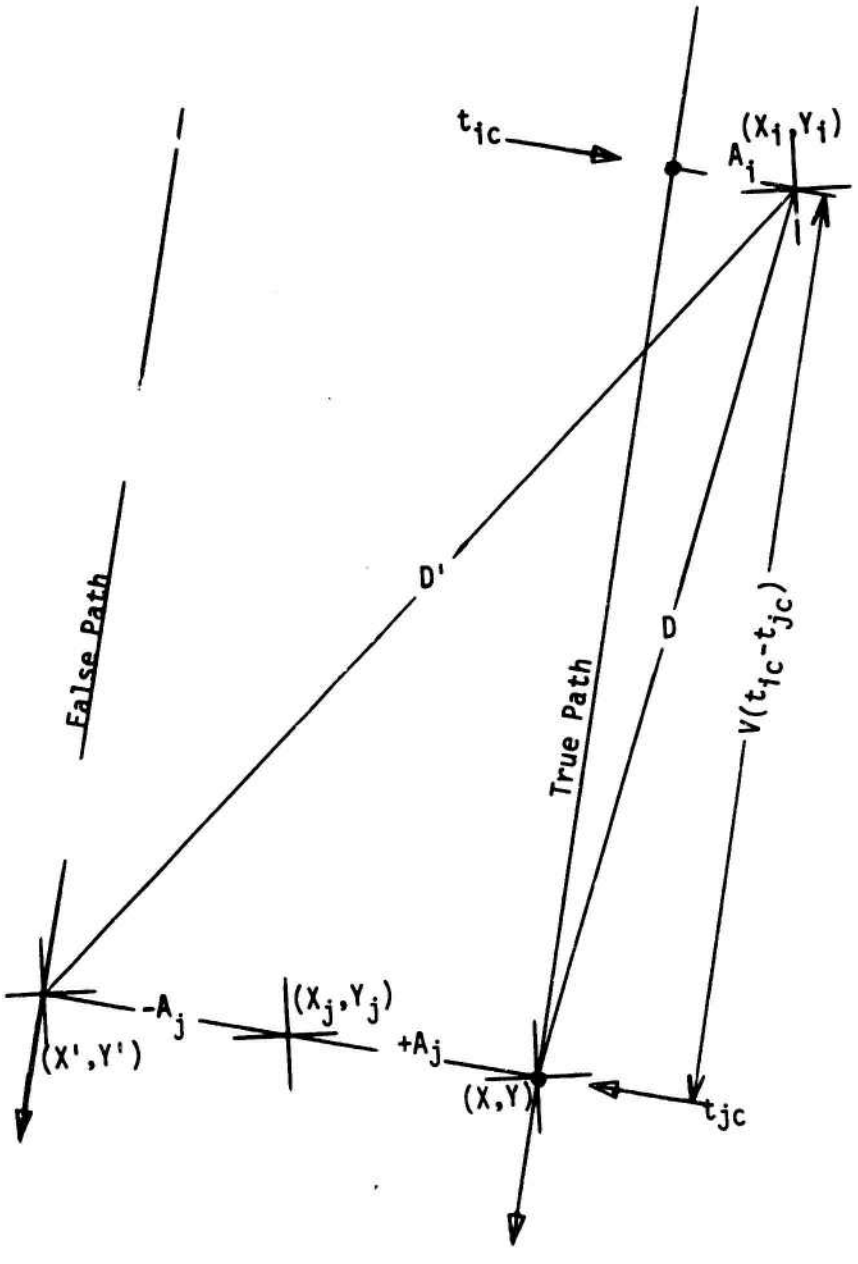
OFFSET TO MULTI-RANGE RELATIONSHIP

FIGURE 7



INTER-DEPENDENCE OF VARIABLES

FIGURE 8



CPA (OFFSET) AMBIGUITY IN TARGET PATH

FIGURE 9

INPUT

Chose:

Target Bearing	5, 10, 20, 30, 45	degrees
Target Speed	5, 10, 15, 20, 25	meters per second
Sensor Spacing	100, 150, 200, 300, 500	meters
Initial Target Offset	.125, .25, .50, .75, 1.25	times spacing

Select 3 σ Value of Error:

CPA Time	0, 2, 4, 6, 8	seconds
Range Ratio	0, 5, 10, 20, 30	percent
Sensor Location	0, 25, 50	meters

OUTPUT

Standard Deviation of the Bearing in Degrees.
Standard Deviation of the Speed in Meters per Second.
Average Coordinates' Error in Meters.
Average Coordinates' Error in Meters projected Sixty Seconds.

RANDOM TARGET SIMULATION

FIGURE 10

EDGEWOOD ARSENAL INCINERATION PROGRAM

William Shulman and William R. Brankowitz
Manufacturing Technology Directorate
US Army Materiel Command
Edgewood Arsenal
Aberdeen Proving Ground, Maryland

ABSTRACT. The Edgewood Arsenal Incineration Program is used to simulate the incineration of military chemicals for process design of incinerator complexes. This computer program is a modified version of the NASA rocket engine performance program. The program minimizes Gibbs Free Energy for a species subject to stoichometrical constraints and yields the stack output. More than 100 chemical species have been considered in some problems. The stack outputs, though in thermodynamic equilibrium, are used to predict actual outputs under real conditions. Specific problems of the newer applications of this program to the Molten Salt incinerator are presented. Statistical methods and estimation techniques of building up the background thermodynamic library are discussed. Specific examples such as the incineration of chemical agent mustard and pesticides are presented.

1. EDGEWOOD ARSENAL INCINERATION PROGRAM. The name of the program which we received early last year was "A Computer Program for the Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks and Chapman-Jougut Detonation". After noticing the program came from the Lewis Research Center of NASA, I was dubious as to what applications we might have for a program originally made for rocket research, since our group is concerned with the demilitarization of chemical warfare agents.

In the way of background, this program was first developed in the early 60's specifically to get information on the combustion products and thermodynamic properties of rocket reactions. The program was built around a library which, by the late 60's, had increased to about 500 fairly simple compounds you would expect to find in the temperature ranges of a rocket exhaust. As time passed, the program was rewritten to take advantage of the advances in computer languages such as the development of Fortran V. For example, the easier data input methods such as the Namelist option were employed. It also had written into it some additional applications such as the Shock routine.

At Edgewood Arsenal in late 1972, we were pondering on how to put this high powered program to use for our purposes. The word combustion was the one which finally caught our eye. We have been charged with the mission of developing processes for the disposal of chemical munitions

which have been declared excess. For many years, burning of certain of these agents has been a tried and true procedure. With the advent of pollution limits and the declaration to excess of certain agents which had never been disposed of in quantity, we were faced with a problem: Could a reasonable model be developed to give us an idea of what to expect in the controlled incineration of a chemical agent? This program has given us this capability.

For a time, we worked on getting the program into shape. We renamed the routine BURN and put it into our program library. A few quick changes to the device numbers, since we use a Univac 1108, produced for us our working program dubbed BURN 2. And along came our first problem - How to get rid of X-S Mustard [$S\text{-}(CH_2CH_2Cl)_2$]. For quite some time, mustard has been burned. It is a fuel oil consistency liquid which has the enviable trait of supporting its own combustion; i.e., the obvious choice of disposal methods is incineration. However with the new pollution laws, we were stuck with a compound which was, in essence, a high sulfur fuel. Analysis had been done on the combustion products at some specific conditions, but our group wished to see what products came off at a variety of conditions of pressure, temperature and mix. The program, in its original form, allows this. Reactants are read in as fuels or oxidizers. Conditions of temperature, pressure and mix are input under one Namelists option - mix being expressed in fuel percent, oxidant to fuel ratio, or two other format choices. Up to 25 combinations of temperature and pressure can be specified for each mix value. The mix values are only limited by the pages you are willing to expend.

Our first runs indicated that we were getting good results, compatible to experimental findings in the known conditions. We soon ran off our runs for a multitude of other conditions to be used as a guide for possibly lowering concentrations of SO_2 and scrubbing the product gasses. In addition, by using a trace routine, we could detect concentrations of out to 35 places past the decimal point mole fraction.

Detecting possible compounds, however, is only as good as your data library. Our first efforts of modification on the BURN Program, then, were directed toward expanding this library. To make our mustard results more significant, we added principally sulphur compounds; at first, mustard itself, the mercaptans, and finally some thio-chlorides.

To add compounds to the library, we found that a regular pattern was followed. First, we would make a literature search for heat capacity, enthalpy, and entropy data on our compound. These are the three building blocks from which the free energy minimization is derived. Here are the equations to which this library data is fitted in the program:

a. $\frac{Cp}{R} = a_1 + a_2T + a_3T^2 + a_4T^3 + a_5T^4$

b. $\frac{H_t^o}{RT} = a_1 + \frac{a_2}{2}T + \frac{a_3}{3}T^2 + \frac{a_4}{4}T^3 + \frac{a_5}{5}T^4 + \frac{a_6}{T}$

c. $\frac{S_t^o}{R} = a_1 \mathrm{Ln}T + a_2T + \frac{a_3}{2}T^2 + \frac{a_4}{3}T^3 + \frac{a_5}{4}T^4 + a_7$

The library is, in fact, a repository of coefficients a_1 to a_7 for each compound in both a high temperature range (1,000°K to 5,000°K) and low temperature range defined from 300°K to 1,000°K. Thus, 14 coefficients are read in on 3 data cards. Other formula, temperature and phase data are stored on a preceding card, the identifying card.

Next, we performed a regression analysis of this data to give us our necessary coefficients. To perform this analysis, we have used the National Bureau of Standards Ommitab II Program. The authors of the NASA Program have now made available a second program known as PAC 2 to process thermodynamic data into cards usable in the BURN library, but we have not yet received a working copy of this program.

When using this program as a combustion routine, we have worked up a number of preliminary guide lines for engineers, which might be helpful to mention now before going further. First, we emphasize that this program should be used only as a guide - as any mathematical model should be. Not to represent firm solutions. Second, it is a thermodynamic guide and does not account for the kinetics of a reaction. This has a two-fold meaning. First, we must assume that the concentrations predicted are those of an absolute equilibrium. This asserts that the combustion reaction has gone thermodynamically to completion. Second, we assume that if the model operates, the reaction will work, <u>though</u> the pressure of some catalyzing force may be necessary to make it occur. We assume that if the model does not operate, the reaction does not occur. Using this last assumption, we have investigated the <u>probabilities</u> of agent reactions with certain chemicals under combustion conditions.

The next problems which occurred forced revisions to the program itself. Under its NASA role, <u>simple</u> compounds had been used for reactants. Thus the reactant cards were formatted in such a way as to have only 5 elements to a compound read. This was fine for mustard, but had to be altered when we wished to simulate the incineration of a nerve agent VX $\left[(CH_3CH_2O)PO(CH_3)(SCH_2CH_2N(C_3H_7)_2)\right]$. As a quick check of the compound shows, VX contains 6 elements.

Thus, to accommodate the VX molecule, changes had to be made to the basic read and write formats dealing with reactant cards. Dimensioning which involved the storage of the compounds basic data also had to be changed. Lastly, changes in the iteration scheme of the data search also had to be made. All total, this numbered slightly over 50 cards revised of the original 3,331. This program was called BURN 3 and was used primarily in an investigation of a persistant phosphorous pollutant which we wished to minimize in our product gasses.

At about this time our library underwent yet another expansion. For the VX problem, several phosphorous compounds were added. A problem with burning a tear agent also added to our growing wealth of compounds. Before long we accounted for some fifty new compounds including organic chlorides, organophosphate, several alcohols and some acids as well. This expansion soon ran into a snag,however. The library, similar to the reactants cards, had been built basically to handle simple products. We were interested, however, in the possibility of any residue of agent which might be left for some thermodynamic reason. Thus compounds much like VX which we wished to add, were faced with a 4 element limit.

We once again set to work to modify the program. On the initial formula-temperature-phase card - the identifying card which I spoke of before - there exists some "dead space" toward the end. Once again, some format, dimension and iteration changes were made. This presented us with a program version BURN 4 which will handle data library compounds of up to 6 elements, by using this available dead space as additional formula data space. This modification has opened up the doors to check for residual molecules of up to six elements which include organophosphorous and pesticide compounds. This version is the current one in use at Edgewood Arsenal by our Division.

By this point in time, several interesting techniques specifically related to using this routine for incineration had been developed. The first of these was the development of sources of data for the compound library. The search for thermodynamic data, particularly if needed for higher temperatures, is frequently long and frustrating. Compounds which seem to be surprisingly simple at times, lack published data even at the lower more reasonable temperatures. The best available collections of data we have found to this time come from the JANAF tables, the "Selected Values of Chemical Thermodynamic Properties" of the National Bureau of Standards, and an article from Chemical Reviews by S. W. Benson et al, entitled "Additivity Rules for Estimation of Thermodynamical Properties". This last mentioned article has data for at least five temperatures for a great many "hard-to-get" compounds. More importantly it contains a method for estimating properties which we have found to be astonishingly accurate. This estimating method is the only way to obtain thermodynamic data in some instances, especially when the compound is very complex or toxic.

Another technique we have developed in the course of using the program is a "simulated pyrolysis". Here, instead of providing a unique oxidant, we have specified the same compound as both fuel and oxidant. Another variation is to consider a compound for the oxidant which will not lend itself to that role. We then run the program as before, setting temperature and pressure at pyrolysis values. We have done this for tear agent and the results were close to those obtained in the laboratory.

During the course of our experience, we have also built up certain typical simulated reactants. One of these was the use of 2 fuel cards in combination to represent Herbicide Orange, the fuel cards simply being representative portions of the compounds known as "2-4 D" and "2-4-5 T". To simulate fuel oil, we have currently been using the formula-plus the thermodynamic values-of decane. In the past, we have used methane alone or with ethane in the proper proportions, to represent natural gas.

One last technique which we developed in using the BURN routine was directly related to the mechanics of the program. The iteration scheme for solving to the minimum free energy has 35 passes allowed. If by the 35th pass convergence is not obtained the program kicks out indicating a reaction is not practical. We found that if we tried to set a temperature initially too high or particularly too low, this occured. By trial and error, we discovered that we were not being thrown out on valid reasons, but that the complexity of our compounds combined with the extremity of our initial conditions caused this. Thus, by setting a normal or average initial condition and stepping down or up to an extremity, say in temperature, the program could be primed like a pump, and the true results could be calculated. Thus, when we want a list of products from 300° to 500°K, we start at 500° and step down to 300°. If a problem is still non-convergent, we assume it does not work thermodynamically.

In the near future, we hope to add the proper compounds to the library to enable us to simulate molten salt incineration. This will simply entail the adding of the appropriate salt compounds to the data library and in working out a proper reactant cards combination to simulate the bed itself. These cards, used in conjunction with the compounds we wish to destroy, will hopefully give us an indication of the composition of the salt bed after the incineration reactions are completed.

In summary, then, several types of incineration processes have been or can be simulated by the use of this program. These can be used to simulate the thermodynamic incineration reaction of nearly any substances, given the proper use of reactants cards and the proper product data for the library. Copies of this program are available by writing to Mrs. Bonnie McBride, its author at the Lewis Research Center. Copies of our library as well as our revisions are available on request from the Commander, Edgewood Arsenal, Attn: SAREA-MT-CP, Aberdeen Proving Ground, MD 21010.

EDGEWOOD ARSENAL POLLUTION ABATEMENT SCRUBBER PROGRAM

William Shulman and William R. Brankowitz
Manufacturing Technology Directorate
US Army Materiel Command
Edgewood Arsenal
Aberdeen Proving Ground, Maryland

ABSTRACT. The Edgewood Arsenal Pollution Abatement Scrubber Program simulates the scrubbing of stack gases for pollution abatement. Scrubbers, quench chambers, and other process equipment can be arranged in any order so as to optimize the process layout. The Algebraic mass balance equations (equation 1) are solved by conversion to differential equations (equation 2) with a convergence constant. The differential equations

$$X_{jn} = \sum_{i=1}^{n-1} A_{ji} X_{ji} \qquad \text{(equation 1)}$$

$$dX_{jn} = k\left[\sum_{i=1}^{n-1} (A_{ji} X_{ji}) - X_{jn}\right] \qquad \text{(equation 2)}$$

are integrated to get the results. A specific example of the use of this program to simulate the scrubbing of stack gases from the incineration of chemical agent mustard is demonstrated. The sample used 150 differential equations representing 15 chemical species output from ten different process units. The computer program is limited to the case of burning in excess air.

EDGEWOOD ARSENAL POLLUTION ABATEMENT SCRUBBER PROGRAM. The Edgewood Arsenal Scrubber Computer Program was written to simulate a scrubbing operation involving N pieces of equipment and M molecular moieties. This Computer Program has not matured to a generalized program, where a few streams can be identified and the Computer Program will setup the proper equation matrix, but its principles have to be tailored to apply from case to case. The mathematical model for the solution of this problem was designed with the following simplifying assumptions:

a. That the material balance algebraic equations can be solved by a series of differential equations, letting the differentials go to zero by successive iterations. 40 iterations were used on the Interdata Model 3 Minicomputer and 100 iterations are used on the Univac 1108.

b. That the various stages remove the same percentage of HCl, CO_2, SO_2, and SO_3 in the reacting with caustic. The material balance corrects the water and caustic concentrations for these reactions. The material

balance considers the equilibrium of $CO_3^=$, $SO_3^=$, and $SO_4^=$ with HCO_3^-, HSO_3^-, and HSO_4^- respectively. Common ion effects are neglected; activity coefficients are assumed unity.

c. That no energy balance need be considered.

d. That all water leaves the various stages as a liquid and that none is in the vapor phase due to humidity, and none is entrained in the gas stream.

e. The caustic must always be in excess, otherwise the program will correct caustic to .004 mole fraction. This was put in to correct caustic in the early iterations so that the equation will not diverge to minus infinity.

f. That the product of the spray dryer is bone-dry material. This is simply accomplished by setting the water concentration to zero.

g. That no consideration is given to the reaction of CO_2 gas from the burning of natural gas with the residual caustic in the spray dryer.

h. That water used is pure and has none of the impurities of industrial water.

i. That caustic used is pure NaOH and does not have carbonate impurities.

j. That O_2 and N_2 are not soluble in water.

k. No solids are entrained in the gas portion. The precipitator of the real problem to be discussed was ignored.

l. That there is only one leaving stream for a particular piece of equipment of one molecular species. Where there are two streams, one must be specified. This must be in order not to have too many unknowns for the given number of equations.

The material balance equation for a piece of equipment is:

$$X_{j,1} + X_{j,2} + X_{j,3} + \cdots X_{j,i-1} - X_{j,i} = 0$$

Where X represents the moles of the chemical under consideration; i represents the streams coming into the jth species. Then, converting to the differential:

$$dX_{j,i} = X_{j,1} + X_{j,2} + X_{j,3} \cdots + X_{j,i-1} - X_{j,i}$$

This equation can present problems diverging if the input numbers are not quite the right values and there is a positive feedback element in the recycle stream, so a correction factor is placed in the equation:

$$dX_{j,i} = k(X_{j,1} + X_{j,2} + X_{j,3} \ldots + X_{j,i-1} - X_{j,i})$$

The dX_i's are integrated at each pass. Our experience has indicated that if k = .7 or less the equations do not diverge, even if these are positive feedback elements. It is obvious that a k greater than .7 would converge faster. An analysis of the feedback loops, a lengthy activity, was not accomplished.

There is a chemical reaction of sodium hydroxide and hydrochloric acid, we simply put in a term in the HCl and NaOH equation indicating a loss of material and in the sodium chloride and water a gain of material. In the formation of carbonate, bicarbonates, sulfites, bisulfites, sulfates, and bisulfates the situation is different.

An equilibrium constant is defined as: $\frac{(H^+)\ (X^-)}{(HX)} = K$

Where H^+ is the hydrogen ion concentration; X^- is the negative ion concentration; HX is the concentration of the combination.

In the case of water: $\frac{(H^+)\ (OH^-)}{(H_2O)} = 1E\text{-}14$

Since the concentration of water is near enough to 100 mole percent so that the hydrogen ion would not change appreciably, the water term is left out. At each pass hydrogen ion is calculated; the (OH) is contributed in the main by sodium hydroxide so that: $(H^-) = 1E\text{-}14 * TN/NaOH$. Using an example of sodium carbonate-bicarbonate system, there are two equilibrium constants:

$$\frac{(H^+)\ (HCO_3^-)}{(CO_2)} = K_1 \qquad \frac{(H^+)\ (CO_3^-)}{(HCO_3)} = K_2$$

The reaction $H_2O + CO_2 \rightarrow H_2CO_3$ was assumed complete for the carbonate-bicarbonate system calculation so that CO_2 was equal to H_2CO_3. Therefore, given CO_2 and H^+ and assuming H_2O as 100 mole percent.

HCO_3^- (as sodium) = $K_1 * CO_2/(H^+)$ and $CO_3^= = K_2 * HCO_3^-/(H^+)$.

The program was written for a scrubber system that was the tail end of an incinerator. The incinerator was simulated by the Edgewood Arsenal Incinerator Simulation Program that was described by Mr. Brankowitz. Results from the incinerator program was used as input data to this program. A process flow sheet of the scrubber system is shown in Figure 1. The incinerator burns Mustard Agent. The off gas from the incinerator is quenched with a solution of fresh make-up caustic and a split stream from the liquid sump. This step essentially brings the temperature down and absorbs the heat from the reaction of caustic with HCl, SO_3, SO_2, and CO_2. The gas stream leaves the quench chamber and is introduced into the scrubber where it is further scrubbed of its HCl, SO_2, SO_3, and CO_2. The liquid used comes from the bottom of the quench chamber. The liquid from the scrubber goes to the

liquid sump where it is pumped to the quench chamber and the spray drier.

An example of the results is shown in Figure 2.

The program was originally programmed on the Interdata Model 3 Minicomputer in Interactive Fortran to be debugged and then converted to the Univac 1108.

The technique which was used to solve the 150 material balance equations, was to convert them to differential equations. By using this technique, if the solution exists, the equations will converge to the solution.

This simulation proved the feasibility of the system. The results do not agree well with the real situation because of the following:

a. The mustard was not pure.

b. The impurities contained sulfur and iron.

In conclusion, this simulation established the feasibility of the scrubbing system and indicated that pollution can be controlled by scrubbing. A copy of the minicomputer program with comments follows.

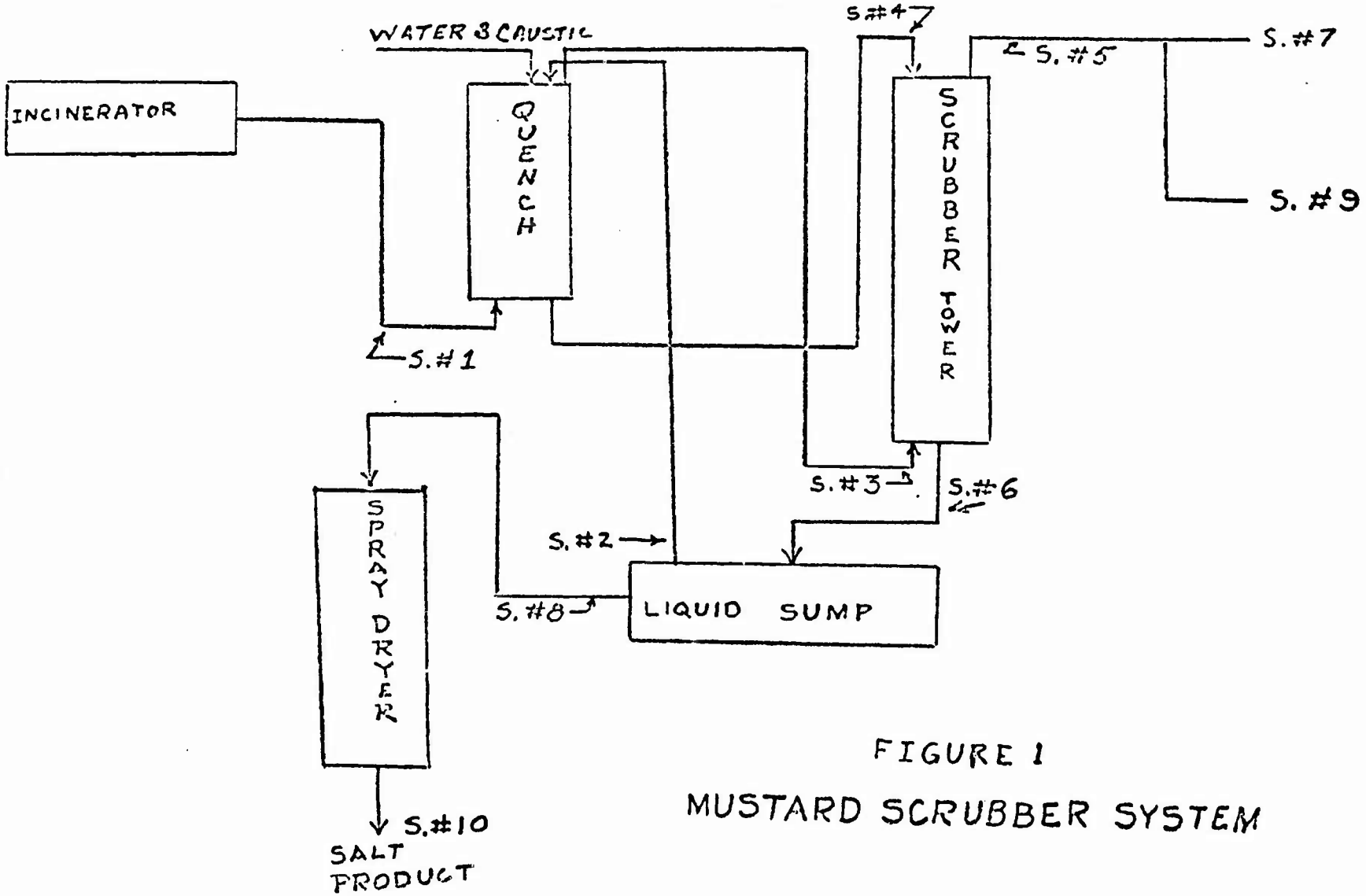


FIGURE 1

MUSTARD SCRUBBER SYSTEM

STATION # 9 - GAS TO STACK

COMPOUND NO.		LBS	%
1	O_2	3.37894	5.00314
2	N_2	63.9677	94.7158
3	CO_2	.118564	.175556
4	HCl	.376908E-01	.558081E-01
5	SO_2	.329725E-01	.488217E-01
6	SO_3	.587336E-03	.869657E-03
7	H_2O	0	0
8	$NaHCO_3$	0	0
9	$NaHSO_3$	0	0
10	$NaHSO_4$	0	0
11	Na_2CO_3	0	0
12	Na_2SO_3	0	0
13	Na_2SO_4	0	0
14	$NaOH$	0	0
15	$NaCl$	0	0

STATION # 10 - BONE DRY SOLID SALT

COMPOUND NO.		LBS @	%
1	O_2	0	0
2	N_2	0	0
3	CO_2	0	0
4	HCl	0	0
5	SO_2	0	0
6	SO_3	0	0
7	H_2O	0	0
8	$NaHCO_3$	.976255	1.71274
9	$NaHSO_3$	.239245E-05	.419733E-05
10	$NaHSO_4$	.986034E-11	.17299E-10
11	Na_2CO_3	28.0186	49.1559
12	Na_2SO_3	7.90374	13.8663
13	Na_2SO_4	.126933	.222692
14	$NaOH$	12.6187	22.1384
15	$NaCl$	7.35513	12.9038

FIGURE 2 - SAMPLE OF RESULTS

```
 1                SUBR AA
 2                DIME C(15,10),D(15,10),C.(7),M(15)
 3                DIME IM(10)
 4                GG=.95
 5                UU=0
 6      C
 7      C         THIS SECTION TAKES IN MOL. WT.
 8      C
 9                DO 1 I=1,15
10                TYPE 'COMP ',I,' MOL WT'
11                ACCE M(I)
12              1 CONT
13                IR=0
14      C
15      C         THIS SECTION DECIDES IF INCINR. CHANGES
16      C
17             79 TYPE 'INCINR CHANGE?
18                ACCE ZZ
19                IF (ZZ) 5,5,92
20             92 TYPE 'INPUT FROM INCINR'
21                DO 93 I=1,15
22                C(I,1)=0
23             93 CONT
24      C
25      C         THIS SECTION TAKES IN THE DATA
26      C
27                DO 3 I=1,7
28                TYPE 'MOLE FRAC. COMP  ,I
29                ACCE C1(I)
30              3 CONT
31                TYPE 'AV. MOL WT,TOTAL LBS.
32                ACCE MW,LB
33                TM(1)=LB/MW
34      C
35      C         THIS SECTION CALCULATES THE NUMBER OF INPUT MOLES
36      C
37                DO 4 I=1,7
38                C(I,1)=C1(I)*TM(1)
39              4 CONT
40      C
41      C         THIS SECTION ZEROS THE VARIABLES
42      C
43              5 DO 6 J=2,10
44                TM(J)=1
45                DO 6 I=1,15
46                C(I,J)=0
47                D(I,J)=0
48              6 CONT
49      C
50      C         THIS SECTION TRANSFERS VALUES FROM ST. 1 TO ST. 3
51      C
52                DO 44 I=1,2
53                C(I,3)=C(I,1)
54                C(I,5)=C(I,1)
```

```
      44  CONT
C
C         THIS INCREMENTS RUN NUMBER
C
          IR=IR+1
C
C         THIS PART CALCULATES THE CAUSTIC INPUT
C         FROM POUNDS OF CAUSTIC
C
          TYPE 'CAUSTIC'
          ACCE NA
          PP=NA/2
          PA=NA/M(14)
          DO 68 IP=1,17
          WRIT UU,' '
      68  CONT
C
C         THIS SECTION TYPES OUT ON TTY #2( KEY UU)
C         ALL THE PERTINANT INFORMATION ABOUT THE
C         RUN. NOTE: THIS FORTRAN DOES NOT RECOGNIZE
C         INTEGER OR REAL VARIABLES.
C
          WRIT UU,'RUN NO. ',IR
          WRIT UU,' '
          WRIT UU,'CUBIC FT/MIN NAT. GAS'
          READ UU,ZZ
          WRIT UU,'LB MUSTARD/MIN'
          READ UU,ZZ
          WRIT UU,'EXCESS MOLAR AIR'
          READ UU,ZZ
          WRIT UU,'GAL OF FRESH 20% CAUSTIC ADDED TO'
          WRIT UU,'QUENCH = ',PP
          WRIT UU,'GAL OF MAKE UP WATER ='
          READ UU,WA
          WRIT UU,'RECYCLE RATE OF RECIRC. TANK IN GPM ='
          READ UU,GM
          GQ=GM*8.33/18
          WRIT UU,'EFFICIENCY OF QUENCH AND SCRUBBER ='
          READ UU,E1,E2
          DO 72 IP=1,33
          WRIT UU,' '
      72  CONT
C
C         NB IS THE AMOUNT OF WATER ENTERING THE PROCESS
C
          NB=(8.333*WA+NA*4)/18
C
C         THIS SECTION SETS THE INITIAL VALUES FOR
C         CAUSTIC AND WATER
C
          C(7,2)=GQ
          C(7,4)=GQ+NB
          C(7,6)=GQ+NB
          C(7,8)=NB
```

```
109                C(7,10)=NB
110                C(14,2)=PA
111                C(14,4)=2*PA
112                C(14,6)=2*PA
113                C(14,8)=PA
114                C(14,10)=PA
115       C
116       C        THE M1 TO M7 VARIABLES ARE MOLES FORMED IN THE
117       C        QUENCH SECTION. THE ARE ADDED TO THEIR RESPECTIVE
118       C        DIFFERENTIAL AND SUBTRACTED FROM CAUSTIC.
119       C
120                M1=E1*C(3,1)
121                M2=E1*C(5,1)
122                M3=E1*C(6,1)
123                M4=E1*C(4,1)
124       C
125       C        THIS SECTION TAKES CARE OF ALL DIFFERENTIALS
126       C        MM IS A DUMMY VARIABLE BECAUSE THIS INTERACTIVE
127       C        FORTRAN HAS NO CARD CONTINUE PROVISION
128       C
129                IA=0
130              7 IA=IA+1
131                MM=C(11,2)-C(8,4)-C(11,4)
132                D(11,4)=MM+M1
133                MM=C(12,2)-C(9,4)-C(12,4)
134                D(12,4)=MM+M2
135                MM=C(13,2)-C(10,4)-C(13,4)
136                D(13,4)=MM+M3
137                D(7,4)=C(7,1)+NB-C(7,4)+C(7,2)
138                D(7,4)=D(7,4)+E1*C(4,1)
139                D(15,4)=C(15,2)-C(15,4)+M4
140       C
141       C        THIS TERM CALCULATES H+ IN STREAM 4
142       C
143                H=1E-14*TM(4)/C(14,4)
144                M5=H*C(11,4)*.226E11
145                D(8,4)=M5+C(8,2)-C(8,4)
146                M6=H*C(12,4)*200000
147                D(9,4)=M6-C(9,4)+C(9,2)
148                M7=H*C(13,4)*50
149                D(10,4)=M7+C(10,2)-C(10,4)
150                MM=2*(M1+M2+M3)+M7+M4+M5
151                D(14,4)=C(14,2)+PA-MM-M6-C(14,4)
152       C
153       C        THIS SECTION CALCULATES STREAMS 3 & 5
154       C
155                D(3,3)=(1-E1)*C(3,1)-C(3,3)
156                D(4,3)=(1-E1)*C(4,1)-C(4,3)
157                D(5,3)=(1-E1)*C(5,1)-C(5,3)
158                D(6,3)=(1-E1)*C(6,1)-C(6,3)
159                D(3,5)=(1-E2)*C(3,3)-C(3,5)
160                D(4,5)=(1-E2)*C(4,3)-C(4,5)
161                D(5,5)=(1-E2)*C(5,3)-C(5,5)
162                D(6,5)=(1-E2)*C(6,3)-C(6,5)
```

```
C
C       THIS SECTION CALCULATES THE DIFFERENTIALS
C       FOR STREAM 6. P1 TO P7 ARE THE MOLES OF
C       MATERIALS FORMED IN THE SCRUBBER AND ADDED
C       TO THEIR RESPECTIVE DIFFERENTIALS AND SUBTRACTED
C       FROM CAUSTIC.
C
            P1=C(3,3)*E2
            P2=C(5,3)*E2
            P4=C(4,3)*E2
            P3=C(6,3)*E2
            D(11,6)=P1-C(8,6)+C(11,4)-C(11,6)
            D(12,6)=P2-C(9,6)-C(12,6)+C(12,4)
            D(13,6)=P3-C(10,6)-C(13,6)+C(13,4)
            D(7,6)=C(4,3)*E2-C(7,6)+C(7,4)
            D(15,6)=C(4,3)*E2-C(15,6)+C(15,4)
C
C       THIS STATEMENT CALCULATES THE H+
C       CONCENTRATION IT IS USED IN CALCULATING
C       P5,P6,P7. THEY ARE CALCULATED USING
C       EQUILIBRIUM CONSTANT FOR HCO3-,HSO3-,HSO4-
C
            H=1E-14*TM(6)/C(14,6)
            P5=H*C(11,6)*.226E11
            D(8,6)=P5-C(8,6)+C(8,4)
            P6=H*C(12,6)*200000
            D(9,6)=P6-C(9,6)+C(9,4)
            P7=H*C(13,6)*50
            D(10,6)=P7-C(10,6)+C(10,4)
            MM=2*(P1+P2+P3)+P4+P5+P6+P7
            D(14,6)=C(14,4)-MM-C(14,6)
C
C       THIS SECTION CALCULATES THE DIFFERENTIALS
C       FOR STAT. 2 & 7 THRU 10 INCL. THE TOTAL WEIGHT
C       OF STREAM 6 IS NECESSARY FOR THE RECYCLE
C       STREAM
C
            QB=0
            DO 8 IQ=1,15
            QB=QB+C(IQ,6)*M(IQ)
          8 CONT
            GA=10*GM/QB
            DO 9 IQ=1,15
            D(IQ,2)=C(IQ,6)*GA-C(IQ,2)
            D(IQ,8)=C(IQ,6)-C(IQ,2)-C(IQ,8)
            D(IQ,10)=C(IQ,8)-C(IQ,10)
            D(IQ,7)=.2024*C(IQ,5)-C(IQ,7)
            D(IQ,9)=C(IQ,5)-C(IQ,7)-C(IQ,9)
          9 CONT
C
C       THIS SECTION CORRECTS THE NUMBER OF MOLES
C       OF THE INDIVIDUAL COMPOUNDS OF EACH STREAM
C       BY THE DIFFERENTIAL TIMES GG A NUMBER
C       LESS THEN ONE TO ENSURE CONVERGENCE
```

```
C
      DO 10 IQ=1,15
      IJ=1
   62 IJ=IJ+1
      C(IQ,IJ)=C(IQ,IJ)+GG*D(IQ,IJ)
      IB=IJ-10
      IF (IB) 62,10,10
   10 CONT
      DO 22 IQ=1,10
      TM(IQ)=0
   22 CONT
C
C     THIS SECTION CALCULATES TOTAL MOLES
C     IN EACH STREAM THE VARIABLE TM(IQ)
C     IS THE TOTAL MOLES IN EACH STREAM
C
      DO 23 JQ=1,15
      IQ=0
   52 IQ=IQ+1
      TM(IQ)=TM(IQ)+C(JQ,IQ)
      IT=IQ-10
      IF (IT) 52,23,23
   23 CONT
C
C     THIS SECTION SETS CAUSTIC AS A POSITIVE
C     NUMBER.
C
      DO 37 IQ=1,10
      IQ=IQ+1
      IF (C(14,IQ)) 36,36,37
   36 C(14,IQ)=.1E-03
   37 CONT
C
C     THIS SECTION COUNTS THE NUMBER OF ITERATIONS
C     THE NUMBER WAS SET TO 40 TO MINIMIZE TIME
C     BUT TO REACH CLOSE TO ZERO DIFFERENTIAL
C
      RT=IA-40
      IF (RT) 7,7,47
C
C     THIS ALLOWS TO TURN OFF PRINTING TTY
C     AT THE ACCE ZZ THE COMPUTER WAITS
C
   47 TYPE 'SET OTHER TT'
C
C     THIS TERM SETS PRODUCT WATER = 0
C
      C(7,10)=0
      ACCE ZZ
      DO 45 I=1,10
      WRIT UU,'STATION # ',I
      QB=0
      DO 24 IQ=1,15
      QB=QB+C(IQ,I)*M(IQ)
```

```
271           24  CONT
272     C
273     C         THIS SECTION PRINTS RESULTS
274     C
275               WRIT UU,'COMPOUND NO.     LBS %'
276               DO 25 IQ=1,15
277               PC=C(IQ,I)*M(IQ)
278               PD=PC*100/QB
279               WRIT UU,IQ,PC,PD
280           25  CONT
281     C
282     C         THIS SECTION GOES TO THE NEXT PAGE
283     C
284               DO 45 HJ =1,16
285               WRIT UU,' '
286           45  CONT
287     C
288     C         THIS SECTION CHECKS FOR MORE RUNS
289     C
290               TYPE 'ANOTHER RUN?'
291               ACCE ZZ
292               IF (ZZ) 91,91,79
293           91  CONT
294               END
```

A COMPUTER-MODELING TECHNIQUE APPLIED TO PRIORITY RANKING OF DEVELOPMENT PROGRAMS

E. H. GAMBLE
US Army Test and Evaluation Command
Aberdeen Proving Ground, Maryland 21005

SUMMARY

A novel application of the subjective Delphi method is found in the selection of the weighting/influence coefficients for a linear modeling study for multiple component development projects with multiple product system applications. The interesting features of the study are the desired individual project relative magnitudes for engineering applied effort treated as the system parameters and their companion best applied percentages of available funding and resources. From preliminary functional flow diagrams per product system, together with the knowledge of that system's required installation/environmental and operational specifications, the weighting coefficients are computed as the linear sum of numbers. Each number (zero to unity) is computed from an average-valued estimate, determined from rating selections for relative complexity for meeting each specification defined under each product system application. The rating selections are made by experienced design/development engineers. The implied importance for each product system is tied to the forecast of expected procurement in numbers and dollars. In addition, a manufacturing and materials processing factor is indicated for each component project to define a relative acquisition difficulty.

The computer results for the designed model are normalized to 100% for total funding allocation and the largest value of ranking effort parameter. An assignment table for effort and fund % allocations results. The sensitivity of the relative project assignments to any design specification may be found. Comparisons of the various product systems with regard to needed resources and complexity can be provided from the model and its study.

I. INTRODUCTION.

In the practical world of competition, a management committed to research and development cannot pursue projects of marginal utility or payoff. The decision to allocate an organization's resources to a given task must be based upon a sound evaluation of the benefits that can result from that particular effort. The decision must be made on the basis of the integrated judgments of the technical, operations, and marketing/user personnel. In order to derive the full value from its investment in research and development, management seeks a position to accurately appraise these independent judgments.

A method has been developed which appears to assist management to make such appraisals and to implement its decisions rapidly and effectively. This method is based on the exploitation and marriage of the optimization techniques used in engineering design; and the techniques used in operations analysis, including the subjective opinion "Delphi" method. It has already been successfully applied to at least one major program of research planning, product development, production, and deployment involving 20 individual development projects applied to somewhat fewer system product lines. This is by no means the number limit for practical application of the systems method developed.

The several methods of SA such as the "Delphi Technique" and the Algorithms of Linear Programming may be combined to solve this difficult problem in engineering planning: the funding required and priority ranking assignments of potential component development projects.

Observations of the numerous applications of the Delphi form for utilizing the subjective opinion of qualified and knowledgeable experts have led us to conclude the need for a careful structuring of the breadth or scope of each question and its numerically rated answer. Good correlation appears to exist in the experts' numerical values for the weighting number for the same narrow scope question. The statistical variance for each answer block may be greatly reduced by this structuring plan.

Using this method, it is possible to assign research and development priorities on the basis of dollar potential within the constraints established by budget allocations, calendar timing, availability of physical resources, and variable deployment factors. The influence of changes in selection criteria (i.e., minimum cost for minimum time; minimum required resources or maximum utilization of a specific factor or group of factors) can be easily translated into terms of resultant changes in funding, priority, and needed resources. It is adaptable for use with either multiproject research and development programs or individual technical operations for productive project planning and control.

Essentially, the method is based on two principles:

1. Recognition and proper weighting of the factors influencing individual judgements made by technical, operations, and user personnel (similar to the Army Delphi method), and

2. Overall product planning with particular attention to how the end result of a given project would affect integrated product lines and their individual components in: (a) a specific business climate, or (b) Army commodity systems for use in a particular tactical field environment.

Mathematical and guided decision-making techniques developed for improvement of component or systems design have proven quite effective. Operations analysis is recognized as the cornerstone of successful manufacturing today. Marketing techniques - particularly in the area of market studies and sales-data forecasts - are, in general, equally reliable and accurate. When properly weighted, each of the influencing factors can be visualized and put in proper perspective in regard to the whole project and defined research and development program.

Each particular technique has been used successfully on many problems. The only novel feature of this study is the viewpoint permitting the utilization of the several methods together for planning purposes.

II. DISCUSSION.

A. A Systems Technique for Engineering Planning.

1. Preparation for Component Definition:

Preparation for the analytical study includes the hardware and software definition of a development project not now available as an "off-the-shelf" item. The best set of design specifications will be determined by the study. The important concern is the setting of a priority ranking and associated % funding allocation consistent with the best estimate of engineering effort required for success in the development and timeliness with the real world constraints. The product systems will apply the development end products as operational components. Because the tactical field environment or installation environment depends heavily on the particular product system, we would require N optimal designs for N systems. Reality forces us to consider a suboptimal design, giving a high adaptability for application. To design to the worse set of specifications and time scales would probably require an excessive engineering effort and funding. We have observed that to ask any creative designer the complexity for a total component development would bring forth an estimate with wide variance from that offered by another capable designer. However, the restriction of each question about the difficulty in meeting only one specification for one application reduces this variance. The utility of the educated subjective opinions of the "Delphi" method is improved by careful structuring of the magnitude and extent of each question proposed.

Let us choose a set of variables $\{x_i\}$, where the subscript i identifies the individual development project. Each x_i is a relative measure of engineering applied effort. Normalization of the value set x_i to the largest of the set will then permit us to identify an ordering or ranking for the N development projects by the descending magnitude of the $\{x_i\}$ (x_i largest) values. The funding allocations will be by the product of $C_i X_i$, where the C_i is a costing number. A weighting or utility matrix with cell values of a_{ijk} describes

much more than a pure preference and serves as a guide to a practical optimized solution of our selection problem. The subscript (j) identifies the complexity-criticality factor (specification) and k the application or product system. We will describe how to select the $\{a_{ijk}\}$, in detail later. First, the C_i could be found several ways, since we are interested in setting $\sum_{i=1}^{N} C_i X_i = 100\%$ of funds allocated. The C_i for computation purposes can be determined as the sum of the cell values of the weighting matrix for the i th column.

The suboptimal solution set we seek is the ranking from the largest to the smallest for the $\{X_i\}^o$ found from a computer solution of our model. The $\{X_i\}^o$ are to be normalized to the largest value. The non-normalized set became a measure of the applied engineering effort to carry out the component development for the i th project. The product of $C_i X_i$ is the companion fund allocation. These $\{C_i X_i\}$ numbers are summed and each normalized to the sum magnitude times 100% to give the percent of recommended allocated funds.

B. Influence of Application Product Systems on Ranking Study.

One philosophy for analysis has included the influence/effect of the environment of the product system which includes the hardware or software as end product of our component developments. To include the multiplicity of each component type and its importance to the system, we must generate a feasible block diagram of each product system with each functional component identified and the representative environmental limits as they influence a given specification. This approach will result in a different set of a_{ijk} values for each K. Also, the multiplicity should be included to properly reflect the importance of a given component development as a part of an overall mix of all Q of the different types of to-be-fielded product systems.

The particular application under study requires the definition of a weighting-influence matrix of N = 17 component

development programs and Q - 14 product systems. A deficiency of N - Q = 3 exists in the number of product systems making use of all of the N different hardware end results of the development projects. To permit the use of the SA matrix technique, we must generate the detail block diagrams for each Kth product system application for the component hardware with care taken to identify any multiplicity of a given ith component.

The 14 constraint inequalities are recognizable in the form of:

$$\sum_{i=1}^{17} a_{ijk}\, x_i \leq w_k \quad = \text{product and investment priority}$$

$$\text{or} \quad \sum_{i=1}^{17} A_{ijk}\, x_i \leq W_k \quad = \text{system engineering complexity}$$

where $(j = 1, 2, \ldots, m)$ is the indicator of the particular specification or time constraint which is influencing the complexity for development of the ith component which must conform to that application requirement. N_{ik} is resources required factor.

$$\text{Where } a_{ijk} = \frac{S_{ik}}{S_t} \cdot P_i^{(k)} \cdot A^*_{ijk} \cdot N_{ik} = \frac{S_{ik}}{S_t} \cdot N_{ik} \cdot A_{ijk}$$

$$A^*_{ijk} = \sum_{1} \left\{ \alpha^{(j)}(\tau_j) + \alpha^{(j)}(x_i) \right\}$$

The $P_i^{(k)}$ symbol is the multiplicity of the ith components used in the K system and $\frac{S_{ik}}{S_t}$ is a cost acquisition/marketing ratio. The $\alpha^{(j)}(\tau_j)$ are criticality numbers in the range of (0 to 1.0) derived from Delphi opinions about development timing and the $\alpha^{(j)}(x_i)$ are complexity companion numbers, again in the range (0 to 1.0). There are more than a dozen j hardware specifications, each requiring a jth weighting value for the ith component when installed into the applicable environment of the kth product system.

The Specification Criteria as Criticality-Complexity Factors

The set of specification criteria used for this study were the following:

$\alpha_{ik}^{(1)}$ (τ_1) - A product system milestone. The value will be near unity when the milestone is a short time away.

$\alpha_{ik}^{(2)}$ (τ_2) - Time is required for expected innovation in materials processing or manufacturing process.

$\alpha_{ik}^{(3)}$ (x_i) - Complexity because of a static design factor or specification (a military design point of operation).

$\alpha_{ik}^{(4)}$ - Because of a dynamic design factor (e.g., transient response).

$\alpha_{ik}^{(5)}$ - Static extreme environmental condition.

$\alpha_{ik}^{(6)}$ - Dynamic or rate of change of environmental conditions.

$\alpha_{ik}^{(7)}$ - RAM requirement.

$\alpha_{ik}^{(8)}$ - Development of an analysis technique prior to design/development, such as software modeling.

$\alpha_{ik}^{(9)}$ - Potential hazard due to foreign materials or substances.

$\alpha_{ik}^{(10)}$ - Excessive energy and power requirements when operational.

$\alpha_{ik}^{(12)}$, $\alpha_{ik}^{(13)}$ - Other pertinent factors.

C. Example for Determination of Representative Matrix Cell Value.

An example has been chosen to demonstrate the iteration-convergence process for finding the optimum convex set of design values.

Let us consider the particular project i = 4 to be the design or development of a hydraulic component. This component will be compatible with other existing components to form a larger subsystem. It is capable (as visualized) of multiple use in a number of product systems. These product systems will have widely different installation environment conditions and different dollar potential. We must establish some arbitrary selection rules which will provide a consistent set of relative values to show the engineering judgment of expected complexity. The value for τ_1 (product goal) will vary with the product. To show an example of the technique, let us set the values for the individual α's at 0, 0.25, 0.50, 0.75, and 1.0. For product K = 1, since our component is needed as a development prototype, we select $\alpha_{4,1}^{(1)}(\tau_1) = 0.50$. The other applicable α's and dominant reasons for the selected values are as follows:

$\alpha_{4,1}^{(3)} = 0.75$ - Difficult specification for static accuracy, for example, position or high force level.

$\alpha_{4,1}^{(4)} = 0.75$ - Small transient time constant, high resonant frequency, and extended functional behavior.

$\alpha_{4,1}(5)$ or $\alpha_{4,1}(6) = 0.75$ - Extended fluid operating temperature, high installation ambient.

$\alpha_{4,1}(7) = 1.00$ - Requirement for very low failure rate.

$\alpha_{4,1}(9) = 0.75$ - High susceptibility to contaminants at extended high and low ambients.

Other α's, such that $\sum_j \alpha_{4,1}(j) = 6.75$.

$$P_4(1) = 7; \; P_4(1) \cdot \sum_j \alpha_{4,1}(j) = 44.5; \; N_{iK} M_{iK} = 0.0029 = S_{ik}/S_t$$

For other K's the individual $\alpha_{4,K}(j)$ will be different. The multiplicity-utilization coefficient, $P_4(K)$, will be different also. For our example, $N_{iK} M_{iK}$ for the same $\underline{K}$ will be $N_K M_K$, not varying with $\underline{i}$.

The N=17, Q=14 Systems Planning Results

The planning model is shown as Figure 1. The final complexity-criticality matrix of all values for our a_{ijk} is given as Table 1.

Table 2 gives a summary of the results for the Delphi value search, the worst value set of summed α coefficients, their normalized $\{Z_i\}$ set by ranking for worst case, and cost coefficients. The computer results for suboptimal $\{X_i\}^o$ and recommended funding allocations are also included.

Evaluation of the computer solution results confirms that the component program priority ranking is strongly influenced by those environmental system factors and the multiplicity of each component's use in the numerous systems. To rank on the

THE ENGINEERING PLANNING MODEL

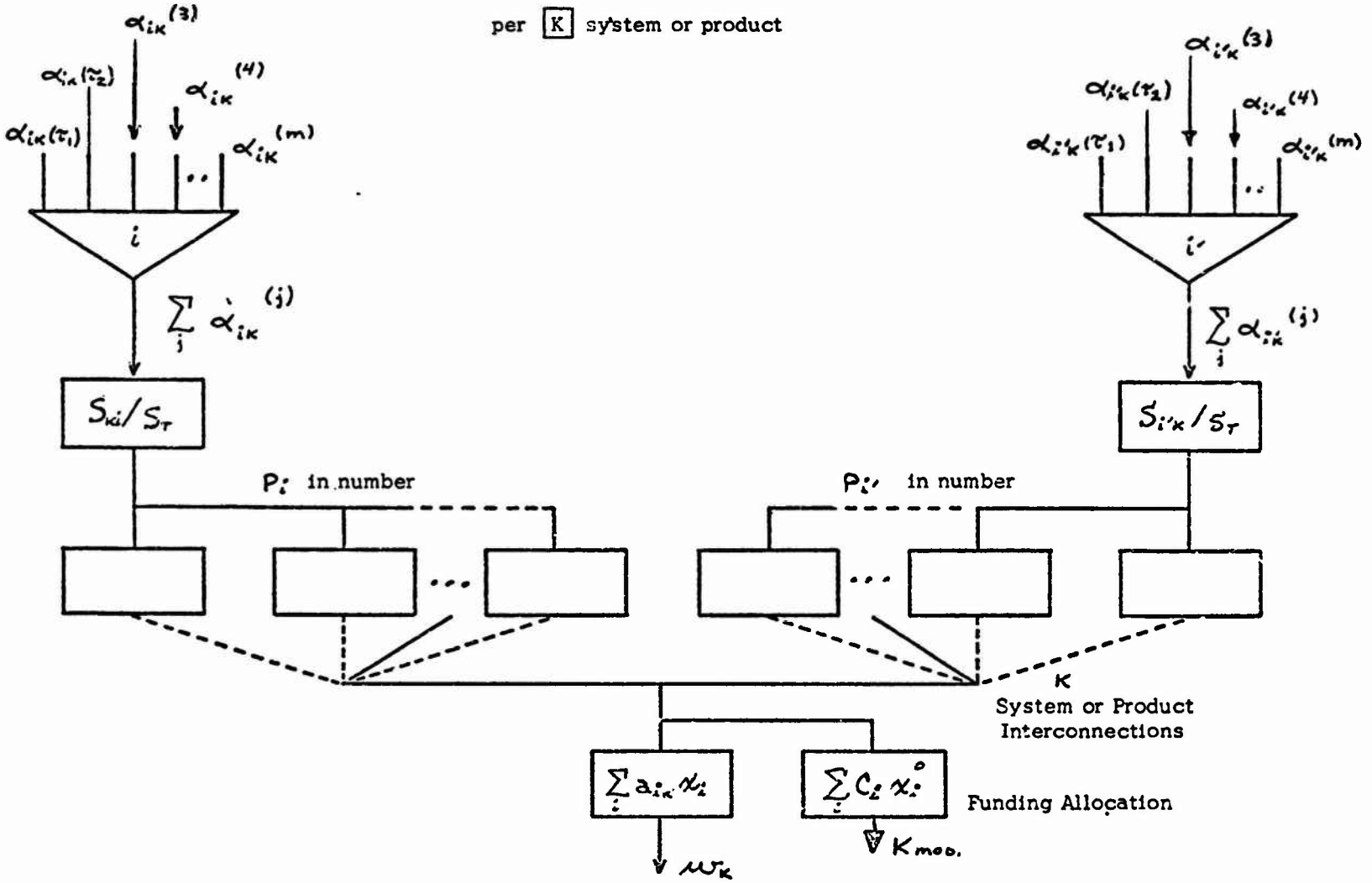


Figure 1

TABLE 1. COMPLEXITY-CRITICALITY MATRIX (SEVENTEEN PROGRAMS)

0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}	B_{16}	B_{17}	(I/O)'	$W_{k\,mod.}$
A_1	3	3	5	2	5	5	10	2	6	1	4	1	1	1	5	1	5	-60	-64.6
A_2	2	2	3	1	3	2	5	1	3	0	3	0	0	0	3	0	0	-28	-30.6
A_3	46	19	30	16	30	41	58	18	40	5	47	10	3	4	29	8	30	-434	-463.9
A_4	28	12	8	17	35	59	39	17	51	7	26	6	4	6	43	0	0	-358	-380.9
A_5	57	26	17	34	73	82	130	35	105	14	55	12	8	11	90	0	0	-749	-791
A_6	42	19	12	25	53	59	95	25	76	10	40	9	6	8	65	0	0	-544	-574.8
A_7	136	48	78	39	78	71	150	30	105	12	71	11	7	10	188	11	0	-1045	-1114.4
A_8	12	5	4	7	15	26	17	7	22	3	11	2	3	2	24	0	0	-160	-167.9
A_9	13	13	15	10	17	19	40	8	25	3	19	3	2	3	25	3	21	-239	-257.9
A_{10}	2	1	1	1	2	3	2	1	3	0	2	0	0	0	3	0	0	-21	-22.3
A_{11}	29	12	8	16	34	57	39	16	47	7	26	6	4	5	54	0	0	-360	-379.5
A_{12}	16	12	20	10	20	18	39	8	25	3	19	3	2	2	30	3	29	-257	-294.3
A_{13}	24	27	43	22	45	40	86	27	54	7	41	6	4	5	54	4	45	-534	-572.8
A_{14}	54	21	42	25	35	31	67	21	42	6	68	6	3	4	50	5	35	-515	-545.7
C_1'	462	220	286	225	445	513	777	216	606	78	432	75	56	61	663	35	163	0	

$m = 17$, $n = 14$, $(m - n)$ = three slack variables

$\Sigma w_{k\,min.} = 530.50 = K_{min}$, $\Sigma W_{k\,min} = 7447$; $\Sigma w_{k\,mod} = 563.7 = K_{mod}$, $\Sigma W_{k\,mod} = 7938.5$

$x_2 = 3x_{12} = 4x_{14}$; $C_2x_2 + C_{12}x_{12} + C_{14}x_{14} = 2C_3x_3$.

TABLE 2 SUMMARY OF RESULTS--ENGINEERING COMPONENT DESIGN (SEVENTEEN PROGRAMS)

Prog. No.	Engineering Complexity-Criticality, $-(\Sigma\alpha)_{max.}$	Z_i (Norm.)	Per Cent of Engineering Effort[a]	Ranking, Z_i	Computer Solutions						
					Funding (Cost) Coefficients	C_iX_i[b]	Program Priority Ranking, x_i			Funding Allocations, C_iX_i	
							Value	Norm.	Order	Value	Order
1	8.75	0.89	8.8	2	46.2	7.81	0.9536	0.3892	8	44.06	5
2	6.25	0.64	6.3	9	22.0	4.99	1.2796	0.5222	5	26.15	9
3	6.25	0.64	6.3	10	28.6	6.42	1.2665	0.5169	6	36.22	7
4	6.25	0.64	6.3	11	22.5	9.78	2.4504	1.000	1	55.13	4
5	5.50	0.57	5.5	14	44.5	6.08	0.8715	0.3557	10	38.78	6
6	4.50	0.46	4.5	16	51.3	14.60	1.6044	0.6548	3	82.31	2
7	6.0	0.62	6.0	13	77.7	20.83	1.5112	0.6167	4	117.42	1
8	4.75	0.49	4.8	15	21.6	2.95	0.7708	0.3146	12	16.65	11
9	6.75	0.70	6.8	6	60.6	3.79	0.3525	0.1439	15	21.36	10
10	7.75	0.80	7.8	5	7.8	1.61	1.1658	0.4758	7	9.09	13
11	9.75	1.00	10.0	1	43.2	5.11	0.6664	0.2720	13	28.79	8
12	6.50	0.67	6.5	7	7.5	0.56	0.4255	0.1736	14	3.19	15
13	1.5	0.16	1.5	17	5.6	0.26	0.2642	0.1078	17	1.48	17
14	6.25	0.64	6.3	12	6.1	0.35	0.3199	0.1306	16	1.95	16
15	8.25	0.85	8.3	3	66.2	10.33	0.8787	0.3586	9	58.26	3
16	6.50	0.67	6.5	8	3.5	1.35	2.1731	0.8868	2	7.61	14
17	8.25	0.85	8.3	4	16.3	2.37	0.8191	0.3343	11	13.35	12
TOTALS		10.20			ΣC_i = 532.3					Σ = 563.79	= $K_{mod.}$
										Σ = 530.50	= $K_{min.}$

(a) Based on Z_i
(b) Based on per cent of engineering effort.

basis of the worst conditions or the equivalent largest value for A^{*}_{ijk} would be giving extremely heavy influence to the engineering complexity without proper regard to the equally important expected utility of the component resulting from the development project. An interesting comparison also can be made by w_K and W_K for the numerous $\underline{K}$ systems. To a large extent, w_K involves a utility number influenced by the total resources needed or product priority ranking. W_K is the engineer's ranking of the systems via the complexity label alone.

The second table 3 summarizes the computer data from the product system and investment priority viewpoints.

Two intermediate parts of the study are worth mentioning. One is the accompanying figure which attempts a simple engineering planning model to provide a road map of the computational process. It will, hopefully, give a clearer explanation than the word of the text.

Table 1 is the weighting matrix of cell values for the totalized complexity-criticality influence coefficients used in the linear programming analysis. The B_i identify the column heads for the 17 component programs and the A_k the product system application.

The $\{X_i\}^{o}$ value set are the suboptimal solution set for the non-normalized decimal percentage of individual applied efforts suggested by the study. The resulting value set must be multiplied by the corresponding C_i used in the programming study of the model. Useful results should be a computed table of the model values for X_i divided by the largest value found, so that the resulting numbers can be placed in a largest to smallest order and can be referenced to a unity value for the highest priority program. A second table of values for the C_iX_i computed for the model divided by the sum of these C_iX_i values times 100% forms a new computed set giving the percentage of allocated funds to be identified for the $\underline{i}$th program.

Table 2 summarized the results of the study with our observation centered upon the component programs. The worst

TABLE 3 SUMMARY OF DATA

Management Total Product or System Ranking				Ratio of w_K	Ratio of w_K mod.	Per Cent of Total	Per Cent of Total Resources Allocation	S_K	S_K/S_T	S_T/S_K	Engineering Complexity-Criticality Ranking			Ratio of W_K	Ratio	Per Cent of Engrg. Effort Allocation(a)	Per Cent of Engrg. Effort Allocation(b)	Comparison of w_K (c)			Evaluation Comments
Prog. No.	w_K mod.	K No.	w_K	w_K max.	w_K mod. max.						K No.	W_K	W_K mod.	W_K max.					W_K	K	
1	111.4	7	104	1.000	1.000	19.4	19.75	23.5	0.156	6.4	14	736	780.8	1.00		9.8	9.87	9.65	1	5	Small return Large invest.
2	79.1	5	74.9	0.72	0.71	14.0	14.02	33.3	0.182	5.5	3	720	770.2	0.98	0.986	9.6	9.73	8.1	2	6	ditto
3	57.5	6	54.0	0.52	0.516	10.1	10.19	23.7	0.132	7.6	13	593	636.0	0.81	0.815	8.0	8.04	11.0	6	4	Good return
4	57.3	13	53.4	0.515	0.514	10.0	10.15	16.5	0.09	11.1	7	660	713.0	0.90	0.881	8.8	8.69	19.4	3	1	ditto
5	54.6	14	51.5	0.495	0.490	9.65	9.68	12.8	0.070	14.3	12	608	641.0	0.82	0.821	8.05	8.10	4.85	5	9	
6	46.4	3	43.4	0.416	0.417	8.1	8.11	11.0	0.06	16.6	1	600	650.0	0.82	0.832	8.05	8.21	1.1	4	12	
7	38.0	11	36.0	0.346	0.341	6.7	6.73	15.8	0.086	11.7	9	571	616.6	0.78	0.790	7.65	7.79	4.45	7	10	
8	38.1	4	36.0	0.346	0.342	6.7	6.75	16.1	0.088	11.4	2	465	498.0	0.63	0.638	6.2	6.30	0.5	8	13	
9	27.4	12	26.0	0.25	0.246	4.85	4.86	7.5	0.041	23.4	11	420	444.6	0.57	0.569	5.6	5.61	6.7	9	7	Good return
10	25.8	9	23.9	0.23	0.232	4.45	4.58	7.7	0.042	23.9	8	420	441.8	0.57	0.566	5.6	5.58	3.0	10	11	
11	16.8	8	16.0	0.154	0.151	3.0	2.98	7.0	0.038	26.3	10	420	440.0	0.57	0.564	5.6	5.56	0.39	11	14	
12	6.5	1	6.0	0.057	0.058	1.1	1.15	1.9	0.010	100.0	5	412	435.0	0.56	0.557	5.5	5.50	14.0	13	2	Large return Low engrg. input
13	3.0	2	2.8	0.027	0.027	0.52	0.53	1.1	0.006	166.0	6	412	437.0	0.56	0.56	5.5	5.53	10.1	12	3	
14	2.2	10	2.1	0.020	0.020	0.39	0.39	0.9	0.005	200.0	4	410	434.3	0.56	0.556	5.5	5.50	6.7	14	8	
TOTALS	$\Sigma w_K = 530.0$				5.064		$S_T = 183.8$					$\Sigma W_K = 7447$			10.135						
	$\Sigma w_{K\,mod} = 563.0$											$\Sigma W_{K\,mod.} = 7938.5$									

(a) Based on system complexity.

(b) Based on product priority.

(c) Total product or system ranking includes cost weighting.

case values for complexity are given as both maximum values and comparison values normalized to that program for which the maximum complexity-ciriticality was determined. A study of the normalized $\{Z_i\}$ value set has major significance to the functional engineering manager. The corresponding ranking value set gives an ordering to the $\{Z_i\}$ value set. In contrast to these worst case value set, the $\{X_i\}^\circ$ normalized values show the heavy weighting influence of the product system applications. This table gives both the computational $\{C_iX_i\}$ funding values, and those values based upon a percentage of engineering effort. Their ordering or ranking numbers may then be compared with the ranking of the Z_i value set.

Table 3 provides the summary of data from the product system observation viewpoint. It provides the data in a form to permit a contrast of the engineering system complexity, $\{W_k\}$ indicators with the management priority values, $\{w_K\}$. The weighting influence of system acquisition numbers and associated dollars causes the differences in the two values sets found. For the management review, comparison figures for the percentage of new development engineering effort based upon system complexity and the percentage based upon a command or business-oriented priority are identified.

III. CONCLUSIONS.

The mixing of a controlled structure "Delphi" technique with a systems engineering design method does appear to be compatible and potentially capable of providing the basis for management decision-making at both the functional engineering and command or general management level. Although such studies can only be viewed as a guide to the detail planning for a total program of development projects, it shows a potential pathway to the identification of the influence effect of resources needed, potential acquisition numbers and dollars for current product system applications of the physical results expected of new development engineering effort. The planner has a broader base for making an analysis than his own limited experience. An additional by-product is that some justification for the ordering and funding of the development projects can be given by the

engineer to the system project managers and financial officers. The potential allocation of each class of needed resources can be based upon an additional source of management information.

IV. RECOMMENDATION.

That serious consideration be given to the utilization of the suggested techniques or some modification of them as a potentially valuable assist for management decision-making at both the functional and command/general management levels.

ECONOMIC, RISK, AND SYSTEMS ANALYSIS OF THE CHEMICAL AGENT/MUNITION DISPOSAL SYSTEM (CAMDS)

John Seigh and Lynn Davis
Plans Office
Analysis Group
Edgewood Arsenal, Maryland

ABSTRACT. In reviewing disposal problems in 1969, the National Academy of Sciences made some general recommendations to the Department of the Army for all future chemical munitions disposal. These recommendations formed the basis of the requirement for a disposal system that could accommodate any of the chemical munitions in the current stockpile. A study was conducted to assess the costs (fixed capital investment, operational, and transportation) associated with several alternative disposal system configurations. These included:

a. Disposal operations conducted at the various chemical storage locations utilizing,

1. A few transportable disposal systems, capable of handling all types of munitions and agents at a given disposal rate, which would serve a given storage location and then, due to its modular construction, could be moved to serve a second or third storage location.

2. A fixed disposal system at each storage location which would be tailored to dispose of the munitions/agents unique to that location and economically optimized in relation to the disposal rate at that location.

3. A mix of fixed and movable disposal systems to serve the nine storage locations.

b. Disposal operations conducted at one, two, or three fixed disposal locations, economically optimized as to disposal rate, assuming that the munitions/agents could be transported to these central disposal locations.

The paper will discuss the computerized model developed to assess the costs associated with the fixed disposal systems, considering the tailoring of the system to a specific stockpile of munitions and scaling up the movable system disposal rate to balance system acquisition costs with operations cost, to determine an optimum cost system.

1. INTRODUCTION AND DESCRIPTION OF THE PROJECT. In reviewing disposal problems in 1969, the National Academy of Sciences made some general recommendations to the Department of the Army for all future chemical munitions disposal. These recommendations formed the basis of the requirement for a disposal system that could accommodate any of the lethal chemical munitions in the current stockpile. An economic, risk, and systems analysis was conducted to investigate alternate disposal system configurations. These included:

a. Disposal operations conducted at the various chemical storage locations utilizing,

1. Four transportable disposal systems, capable of handling all types of munitions and agents at a given disposal rate, which would serve a given storage location and then, due to its modular construction, could be moved to serve a second or third storage location.

2. A fixed disposal system at each storage location which would be tailored to dispose of the munitions/agents unique to that location and economically optimized in relation to the disposal rate at that location.

3. A mix of fixed and movable disposal systems to serve the nine storage locations.

b. Disposal operations conducted at one, two, or three fixed disposal locations, economically optimized as to disposal rate, assuming that the munitions/agents could be transported to these central disposal locations.

A computerized model was developed to assess the costs associated with the fixed disposal systems and the optimization of these costs. In addition, a second model developed to assess the transport of the chemical munitions or agents to optimized fixed disposal facilities will briefly be discussed.

The Chemical Agent Munition Disposal System (CAMDS) is presently being developed at Edgewood Arsenal. The CAMDS is a modular system made up of some 40 building blocks or units and is capable of being moved to several locations for disposal activities. Only the concrete pads and utility supply are fixed at a disposal site. The components are technically complex and require extensive controls to meet the stringent containment, safety, and air quality requirements. It is envisioned that four such movable systems could be built to serve the nine chemical storage locations. Each system is required to be "universal," i.e. capable of handling approximately 11 munitions - rockets, land mines, bombs, spray tank, with a variety of agent fills - H, GB, VX. Some munitions contain explosive components and some do not. The various combinations of munition carriers, agents, and explosive containment represents some 30 different munition configurations, each with an associated disposal rate based on the capability of the movable system. These disposal rates are constrained due to the transportability requirement of the system and were subsequently found to be generally below those rates which would be economically optimum. That is, the subsequent operations costs associated with these four movable systems far outweighed the system acquisition costs. It was postulated that nine individual disposal systems, tailored to the storage locations, i.e., munition stockpiles they were to serve, and operating at higher disposal rates, might be more economical then the four movable systems. Hence, efforts were initiated to develop a model which would scale up the capability of the movable disposal system and determine an optimum cost system, i.e., the point where the sum of the system acquisition cost and system operations cost is a minimum, as shown in Figure 1.

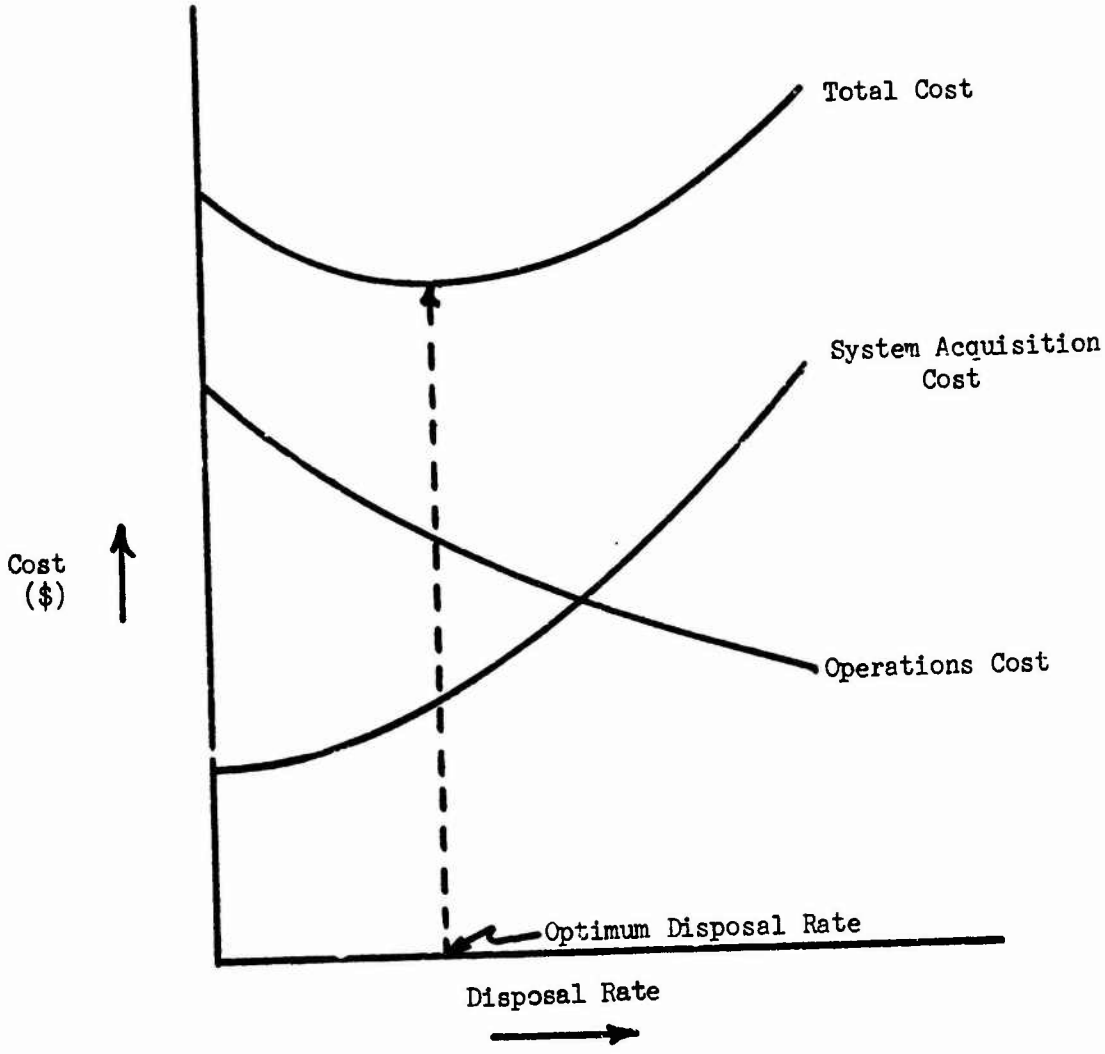

Total Cost
System Acquisition Cost
Operations Cost
Cost ($)
Optimum Disposal Rate
Disposal Rate

FIGURE 1

The computerized model developed is capable of providing elements of cost associated with a system (acquisition, engineering, operations labor, replacement parts, materials and utilities required during operations, etc) for any predetermined stockpile composition, quantity, and disposal rate. The optimum disposal rates determined are those which exhibit the lowest total cost of the sum of the various cost elements.

It was realized early that an infinite number of disposal rates for each of the 30 munition types existed, and the problem had to be constrained. Hence, the munitions were segregated into four broad categories, each having many common characteristics.

These are:

Category A - Rockets & Mines
B - Projectiles with explosives
C - Projectiles without explosives
D - Bulk items (bombs, spray tanks, ton containers of agent, etc)

Subcategories were then used to distinguish major differences among munitions within a category.

A basic disposal rate was defined as the rate of disposal associated with the CAMDS or movable system. This basic rate is referred to as a 1 1 1 1 disposal rate combination system where each digit is associated to the A, B, C, and D munition categories respectively. In reality, the basic rate may represent 400 rockets, 575 mines, or 5 spray tanks per day. Varying the disposal rate, however, was limited to the four rate combinations associated with the munition categories and these rate categories were varied by integer multiples.

The computer program is capable of providing a miriad of information relative to a system at any specified disposal rate. Included are the scaling and costs of individual building blocks, the labor required for operation in numbers of people and their cost, the cost of materials and utilities during operation, an estimate of the cost of replacement equipments during operation, and the time of operation associated with portions of the stockpile or the total stockpile.

As was mentioned previously, the CAMDS system is composed of 42 building blocks or modules. This system, and its associated costs and capabilities, served as the basis upon which the fixed systems were determined. Although the building blocks of the fixed systems were no longer required to be movable, the modular concept was retained. For the fixed system, a building block or module is related to a function rather than an operational unit. A listing of the building blocks which make up a universal system is shown in Table 1 (next page).

TABLE 1

BUILDING BLOCKS

1. Unpack Area
2. Explosive Containment Cubicle
3. Deactivation Furnace
4. Deactivation Furnace Scrubber
5. Metal Parts Furnace
6. Punch, Drain & Saw
7. Dunnage Incinerator
8. Utilities
9. Utility Module
10. Control Module
11. Control Point
12. Personnel Support Complex
13. Agent Destruction System
14. Explosive Treatment System
15. Saw, Dump & Probe
16. Burster Size Reduction
17. Thaw Station
18. Pull, Drain & Rinse
19. Central Decon Supply
20. Projectile Disassembly Facility
21. Bulk Item Facility
22. Materials Handling Equipment
23. Filters
24. 4.2 In. Mortar
25. M23 Land Mine
26. Piping
27. Electrical
28. Scale Model
29. Perimeter Monitoring
30. Closed Circuit TV
31. Communications
32. Chemical Laboratory
33. Detectors
34. Technical Data Package
35. Systemization
36. Training
37. Repair Parts
38. Systems Management & Planning
39. Other Edgewood Support
40. Site Preparation
41. Bulk Agent Facility
42. Military Construction Army

The stockpile at a given disposal location may permit the omission of several of these building blocks from the system developed. For example, a location at which no mines or mortars are stored, would not require building blocks 24 and 25 to be included in that system. Likewise, the portion of cost associated with the transportability of the module need not be included in a fixed installation and these costs were eliminated from the basic cost data.

2. THE COMPUTER MODEL. The program was written to reflect the ideas, opinions, and estimates of engineers familiar with the design of such a facility. There were frequent meetings and discussions resulting in rewriting, adding, and eliminating portions of it.

The objective sought was to provide some insight as to what size facility (or rate of munition-chemical agent disposal) would be most economical. Therefore, the model must generate a total cost of first acquiring or building a facility tailored to certain munition types found at a particular site, and, secondly, operating that facility for the time required to dispose of all of each type found at that site. This is a good point at which to emphasize that not all cost elements have been included in the model. Generally, only those which would be related to disposal rate were considered for the purpose of analysis.

A flow chart of operations performed in the computerized model is illustrated in Figure 2. The computer program is responsive to three site-related inputs:

First it must be given the specific munition types which must be processed through it, that is, those which are peculiar to the disposal location or site (the introduction specified some 30 different types or unique combinations of the carrier munitions with and without explosive components and their chemical agent fills).

It also, quite naturally, requires the inventory of each of the types.

The last input that requires any explanation is the rate multiples. A number of multiples is provided for each of the four munition categories, unless the particular site has no munitions of a category; then the number of multiples for that category is set to zero. Only integer multiples were used mainly because many of the engineers' estimates were based on the CAMDS batch processing components and little confidence could be taken in estimates for the cost of a component which would operate at some slower rate than the CAMDS. The multiples for the non-zero categories are chosen by making a cursory examination of the inventory. An example of this process is shown for an hypothetical storage site:

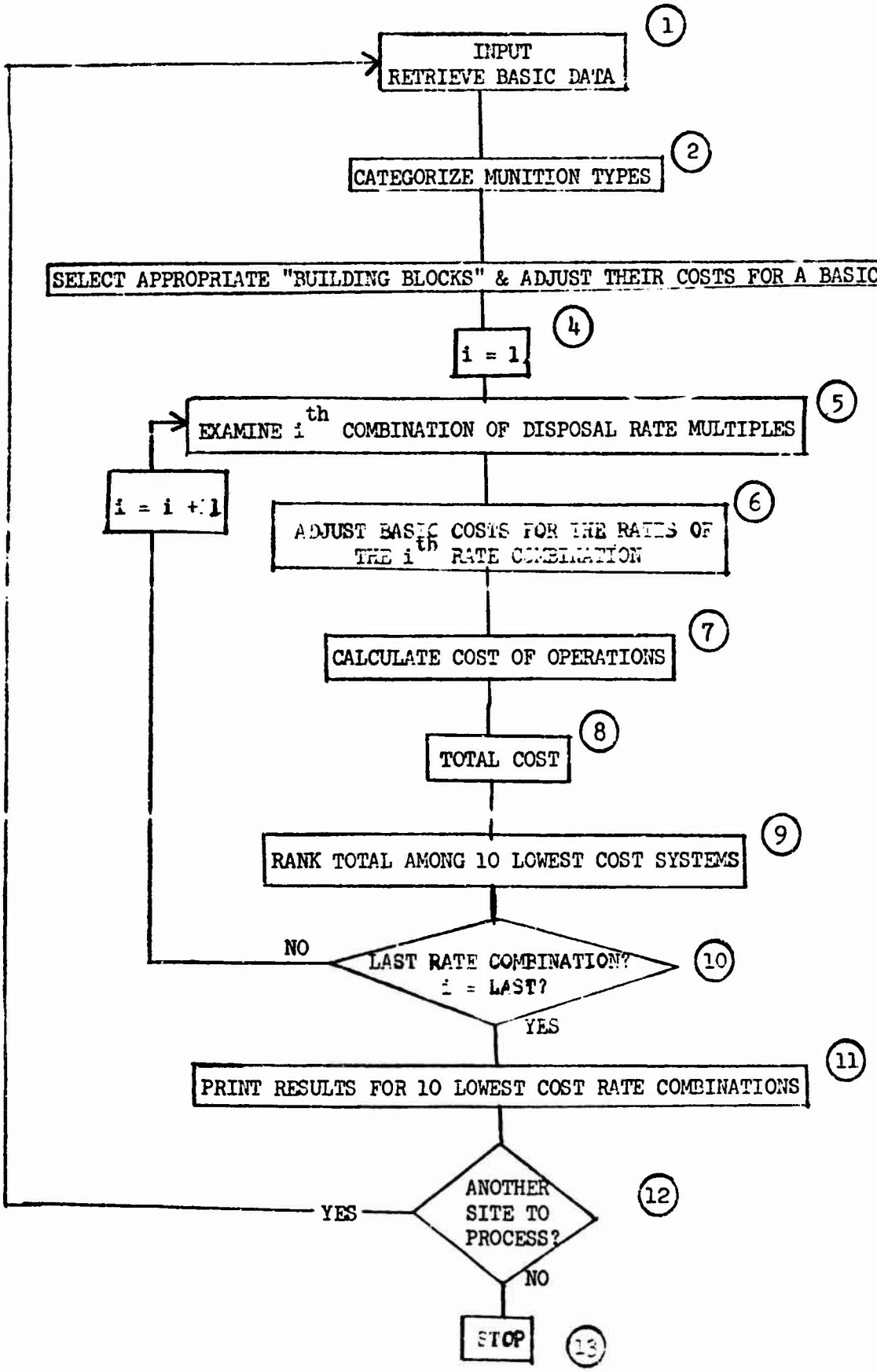

1
INPUT
RETRIEVE BASIC DATA
2
CATEGORIZE MUNITION TYPES
SELECT APPROPRIATE "BUILDING BLOCKS" & ADJUST THEIR COSTS FOR A BASIC SYSTEM
4
i = 1
5
EXAMINE i th COMBINATION OF DISPOSAL RATE MULTIPLES
i = i + 1
6
ADJUST BASIC COSTS FOR THE RATES OF THE i th RATE COMBINATION
7
CALCULATE COST OF OPERATIONS
8
TOTAL COST
9
RANK TOTAL AMONG 10 LOWEST COST SYSTEMS
NO
LAST RATE COMBINATION?
i = LAST?
10
YES
11
PRINT RESULTS FOR 10 LOWEST COST RATE COMBINATIONS
YES
ANOTHER SITE TO PROCESS?
12
NO
STOP
13

FIGURE 2

SITE: EAST PIGEONTOE ARMY DEPOT

General Description of EPAD's inventory:

Category A - large numbers of rocket and mines
Category B - no items (explosive projectiles)
Category C - a few items (non-explosive projectiles)
Category D - significant numbers of bulk items

The analyst now might very well choose to examine the following rate multiples based upon the above information:

Category A - 3,4,5,6,7,8,9,10	(total of 8)
Category B -	(total of 0)
Category C - 1,2,3,4	(total of 4)
Category D - 3,5,6,7,8,9	(total of 6)

With these inputs the program will examine the total cost for each of 192 systems (the number of combinations of rate multiples for "non-zero" categories is the product of the number of multiples in each "non-zero" category; 8 x 4 x 6 = 192).

Inspection of the results should make it obvious whether or not the rate multiples chosen include an inflection point for total cost when considering the trade-off between acquisition cost and operations cost.

Other inputs include: Monthly depot storage cost; replacement parts rate, a percentage; average hourly wage rate; and the time, in days, for systemization which is a period for system checkout and operator training prior to full scale operation.

The program next receives the so-called "basic data" from mass storage- this is all the data that is not site-related - it was updated as the engineers refined their ideas and estimates. The parameters include "Basic System" or CAMDS figures for:

a. Building Block or actual hardware costs.

b. Building Block Engineering Support Costs.

c. Labor or staffing requirements for each munition type.

Also included are the cost-rate relationships, the required physical data for each munition type (mass of metal, of agent, and disposal rate in items/days), and finally, the "partial system" factors which will be explained next.

The munition types which were input in step 1 are now categorized. These categories then determine which building blocks are required and also determine any basic adjustments to be made to those required. The basic adjustments are made by the assignment of factors ranging between 0 & 1 for each category and for each building block resulting in a 42 x 4 matrix of factors. An example of the factors and how they are used for one of the building blocks is shown in Figure 3.

The cost of BB#20, the Projectile Dissassembly Facility, to process all munition categories is estimated at $60K. To process just any one category the factors in Figure 3 (.3, .85, .85, and .6) were developed to reflect the estimated savings in not having to build in the specifications for all categories. For a site tailored to specific categories it was decided to sum the factors or fractional costs of all required categories until that sum exceeded 100% of the basic system cost. Since the given stockpile consists of only categories A & D, the basic BB#20 costs 90% of the CAMDS cost or $54K. In our example, if category C munitions also required processing, the sums would exceed 100% of $60K and would be reduced to $60K. The assumption made is that the fractional portions for each category are mutually exclusive; an assumption obviously not generally true; but, where it was felt the difference from true cost was significant, this general rule was not used.

The above description is the general rule for tailoring the CAMDS system to specific munition types. There are several exceptions to this rule, of which BB#5, the Metal Parts Furnace, is an example. This is the system component that processes the munition hardware after it has been drained of chemical agent but retains a residual contamination. The basic or CAMDS cost varies not necessarily with category, but rather with munition size and agent type. The rate of metal through-put in lbs/day is interrelated with munition size and agent type to influence the scaling up cost adjustment. The logic became so involved for both the basic and the scaling up adjustments that a separate sub-routine was written to accommodate the problem. So the metal parts furnace has provided us the extreme example of diversion from the general rule.

The heart of the program begins in steps 4 and 5 of the flow chart (Figure 2). A combination of rate multiples is selected and all costs for the tailored, scaled-up system and its operation to completion are found in the loop from step 5 through step 10. The tailored basic costs found in step 3 are now subjected to revision in step 6 so they will correspond to the rate multiple combination as follows:

Generally,

a. From step 3 each category associated with each building block is examined to determine whether or not it requires that building block (a zero factor indicates no need).

FIGURE 3

BB#20 - PROJECTILE DISASSEMBLY FACILITY ($60K)

A	B	C	D
.3	.85	.85	.6
($18K)	($51K)	($51K)	($36K)

GIVEN: A SITE WITH MUNITIONS IN, SAY, CATEGORIES A & D ONLY

FIND: COST OF BASIC RATE SYSTEM TAILORED TO THE GIVEN SITE:

A	B	C	D		
.3	0	0	.6	=	.9
$18K	0	0	$36K	=	$54K

b. Now the rate multiples associated only with those categories requiring a building block are examines to find the largest.

c. This largest required multiple is applied to that basic cost found in step 3 of the flow chart in one of the following three ways:

1. Unchanged or fixed (cost is independent of rate).

2. Linear (the largest required multiple is simply multiplied by the step 3 basic cost).

3. Exponential (cost change with rate less sensitive than with linear-largest required multiple is raised to some power between 0 & 1 and then multiplied by step 3 basic cost.

So returning to East Pigeontoe we can find the final cost of, say, BB#22 using one of the 192 rate combinations, say, (4, 0, 3, 8).

a. We first note that it follows the general rule; the basic data supplies that information to the program.

b. Then, from the basic data, we find that only categories A, B, & C require BB#22, i.e., bulk items, Category D, do not require it, and the combination of required rate multiples becomes (4, 0, 3, 0).

c. Now the maximum of the required rate multiples is isolated; MAX (4, 0, 3, 0) = 4.

d. From the basic data the factors again for the required multiples only are summed; .5 + 0 + .3 + 0 = .8.

e. This factor is applied to the CAMDS cost and the basic system cost peculiar to East Pigeontoe is \$154K; (\$193K) (.8) = \$154K.

f. Finally, the basic data supplies the information that BB#22 is to be scaled exponentially with an exponent of .6. So, maximum required rate multiple found in c. above is raised to the .6 power, and the result is applied to the basic East Pigeontoe BB#22 cost for the final cost; $(4)^{.6}$ (154) = \$355K.

All building block costs which follow the general rule are found in this fashion. Some follow the rule with slight modifications and a few such as the Metal Parts Furnance discussed earlier, require a vastly different method.

The other elements of acquisition cost is Engineering Support. Engineering Support costs were estimated for each building block in a way similar to, but less complicated than for the hardware.

Moving on to step 7 of the flow chart (Figure 2), the second and final major cost element is found. The cost of operations is primarily the cost of the labor force required. The size of the labor force is determined by beginning with engineering estimates of the total number required to operate a basic (CAMDS) plant for <u>each</u> munition type.

A standard engineering practice used to scale up production is to fracture the total staff required for a known or basic system into components similar to the way building blocks costs were treated, i.e., fixed, linear, and exponential portions.

F_i - fixed number (independent of rate)
L_i - linear portion
E_i - exponential portion

The three components are then subjected to the appropriate rate multiple, M_i, and summed to give the revised total for a munition being processed at a specific rate.

$$T_i = 1.1 \left(F_i + (L_i)(M_i) + (E_i)(M_i)^{.6}\right)$$

The result is a total staff, T_i, to operate the plant on a four shift/day, 5 day/week, basis for munition type i at rate multiple, M_i. A 10% factor is included to provide for scheduled and unscheduled absences such as leave, sickness, etc. A further refinement was necessary to provide a weekend non-operating staff. An estimated 47 positions are required for these two days - this number is independent of disposal rate. That is equivalent to 19 men working a 5-day week. However, because of scheduling problems which occur it was decided to add 25 positions.

$$T_i' = T_i + 25$$

Now that we have the size of the staff, the cost of labor is found in two separate activities;

First, the cost of the actual operation of the plant at site j is found by examining each munition type i.

1. Actual Demil/Disposal Process Munition <u>i</u> at site <u>j</u>

$$c_{ij} = 40\, T_i' W_j \frac{(1.25\, I_{ij})}{5 R_i M_i}$$

where,
c_{ij} - total disposal cost associated with munition i at j.
40 - number of working hours/employee/week.
T_i' - staff required for munition i at the disposal rate for the category of munition i.
W_j - average hourly wage rate at site j.
1.25- provides for 25% unscheduled downtime.

I_{ij} - the inventory of munition i at site j.
5 - number of operating days/week.
R_i - the basic disposal rate (items/day) for munition type i.
M_i - the rate multiple being considered for the category containing munition type i.

The total labor cost for operations at site j, then, is simply the sum of costs for each munition type.

$$C_j = \sum_{i=1}^{N_j} c_{ij}$$

where,
N_j is the number of different munition types at site j.

The second element of labor cost occurs during the changeover process when the facility has completed operation on one type and must prepare for the next - a period of 2, 3, or 4 weeks depending upon the two munition types involved in changeover m. For N munition types there are N-1 changeovers. The cost of labor, then, for changeover m at site j is shown by this expression.

2. Changeover labor cost

for changeover m:

$k_{mj} = 40\ t_m T_m W_j$
$m = 1, 2, \cdots, N-1$
t_m = time in weeks for changeover between munition types i and i + 1.
T_m' = larger of staffs T_i' & $T_i' + 1$

Summing over m gives the total cost of all changeovers at site j.

$$K_j = \sum_{M=1}^{N-1} k_{mj}$$

Finally summing the operations and changeover labor totals gives the total labor cost.

Total Labor Cost at Site j = $C_j + K_j$

Similar procedures were used to get costs for materials and utilities and the depot storage costs. The final operations cost element is for replacement parts - this was estimated by taking a percentage of the total hardware cost as an annual replacement parts rate.

Finally in step 8 of the flow chart (Figure 2) we sum the acquisition and operations costs to get a total cost for one specific combination of rate multiples.

It is then ranked among the ten lowest cost rate combinations or rejected as higher than the tenth lowest in step 9.

The looping (steps 5 through 10) terminates when the final rate combination has been examined and the results can be printed in various detail using write options provided.

Step 12 provides for stacking as many cases (sites) as needed.

No results will be provided here, however, we will state that the program was used to perform limited sensitivity analysis on the basic building block costs and basic engineering support costs; also on average wage rates and replacement parts percentage. This proved useful not only in assessing their effect on total cost but also the optimum rate combination.

A second program, written by Mr. Philip Robinson, provided results necessary in the phase III analysis mentioned previously; the relocation of the munitions to one or more of the 9 sites. A preliminary decision was made to consider just 1, 2, or 3 predetermined disposal sites and the remaining 8, 7, or 6 respectively would then contribute their stocks to the disposal site(s) and become "feeder" sites. A matrix of transportation costs for shipping all munitions from each feeder site to each disposal site was developed and used to get a total shipping cost for any distribution of the feeder sites to the disposal sites. There was no attempt made to divide the stock at one feeder site and send the parts to more than one disposal site. Optimizing the rate multiples for the single site case was a simple matter of combining all the inventories of the 9 sites, and subjecting this total to the optimization program. However, for the 2 and 3 disposal site cases, there are respectively, 128 and 729 ways to distribute the feeder sites to the disposal sites. For each of these combinations an optimization of rates was necessary. Mr. Robinson's program uses the first program with several required modifications to optimize the rates, then calculate the transportation costs, ranks that feeder site combination for lowest cost, establishes another feeder site combination and continues until all combinations have been examined and compared for lowest cost. It prints the five lowest cost configurations, the optimum rate multiples, and transportation costs broken down for each disposal site.

Finally, it should be pointed out that the more standard O.R. techniques for optimization such as mathematical programming and network models were investigated for applicability before developing this specific model.

The models discussed were developed to support an economic analysis of alternative chemical disposal options based on the assumption that all existing toxic stockpiles would be demilitarized over an extended period of time. It is expected that the analysis will be updated periodically as better cost estimates are available and studies relative to stockpile retention are completed. This paper was presented to illustrate a use of computerized methods in cost estimating and cost analysis.

FORECAST OF SCHEDULE/COST STATUS UTILIZING COST PERFORMANCE REPORTS OF THE COST/SCHEDULE CONTROL SYSTEMS CRITERIA: A BAYESIAN APPROACH

M. Zaki El-Sabban
Directorate for Plans and Analysis System Analysis Division
U. S. Army Aviation Systems Command
St. Louis, Missouri

ABSTRACT

This report presents a Bayesian approach to a forecasting technique useful in projecting the future cost and schedule of work breakdown structure (WBS) items in Department of Defense contracts. The technique utilizes the data supplied in the cost performance reports (CPR) of the cost/schedule control systems criteria (C/SCSC). The forecast data are invaluable to the project manager supervising the contract, who might thereby avert costly schedule/cost program overruns. The advantages of this method are discussed in the present report and a solved example is hereby given.

FOREWORD

The present study was initiated by Mr. John W. Hollis, Chief, Systems Analysis Division, whose continued interest throughout this work is hereby acknowledged.

This is Technical Report 73-1 of the U. S. Army System Command.

I. INTRODUCTION

The adoption of the Laird-Packard principle of closer ties with the contractor and more efficient supervision of the project operation throughout the life of the contract, created a new concept - that of establishing standards or criteria which enunciate the capabilities of a good cost/schedule management system, but leave the details of how to achieve these capabilities to the contractor. These standards are known as the cost/schedule control systems criteria (C/SCSC) and are contained in the Department of Defense Instruction 7000.2 (DODI 7000.2) . Generally, the C/SCSC require that the contractor's activities be integrated and performed in a formal, disciplined fashion, which will allow a follow-up of his contractual progress. The contractor is also required to periodically provide the program manager with work breakdown structure (WBS) summary data that allows assessing the contractual adherence to the approved cost and schedule plan of action for each of the items in the WBS, especially the cost and schedule of the project at completion. The cost performance report (CPR), furnished about once a month by the contractor for this purpose, relates the costs incurred to date to the budgeted cost of the work actually performed, as well as to that of the work originally scheduled. Variance of the cost and schedule of each item from those originally planned, are also reported. The data reported in the CPR are very valuable to the project manager because it keeps him updated about the progress of the contract. What is more important, however, is to be able to use these data to gain insight in the future status of the program. Stated otherwise, the project manager needs a forecasting tool. Such tool would enable him to predict cost/schedule problem areas that might require his immediate attention. Such vital information might be so valuable as to make it possible to avert costly schedule/cost program overruns. There are several possible forecasting techniques. Of these, two familiar ones are the discounted least squares technique and the time series. The former technique would involve extrapolation of the least squares equation, far beyond the available data range, a very risky procedure that might well lead to erroneous conclusions. The time series method is technically superior and dependable but much more elaborate.

The continuous influx of the monthly status reports (CPR) prompted the investigation of the Bayesian statistical approach,

---

[1] Department of Defense Instruction DODI 7000.2, as Appendix E to Army Regualtion AR37-200, August 1968.

which calculates a posterior probability from an assumed prior probability. The present developed Bayes technique forecasts the expected cost/schedule at a future point in terms of the current data, as well as the variances to those forecast values.

II. METHODOLOGY

Bayes' theorem states that[2]:

$$p(\theta|\theta_0) = \frac{p(\theta)\, p(\theta_0|\theta)}{p(\theta_0)}$$

$$\propto p(\theta)\, p(\theta_0|\theta)$$

(since $p(\theta_0)$ is independent of θ and equals $\int p(\theta)p(\theta_0|\theta)\, d\theta$) $p(\theta|\theta_0)$ is the posterior pdf for the parameter vector θ, given the sample information θ_0; $p(\theta)$ is the prior pdf, for the parameter vector θ; and $p(\theta_0|\theta)$, viewed as a function of θ is the likelihood function. The following is a possible manipulation of Bayes' theorem to apply to the C/SCSC problem:

Assume that an item cost/schedule at some point "o" along the project be normally distributed with mean μ and variance σ_0^2. If the estimated cost/schedule at this point is μ_0, then the likelihood function is

$$p(\mu_0|\mu) = \frac{1}{\sqrt{2\pi}\,\sigma_0} \exp\left[\frac{-1}{2\sigma_0^2}(\mu_0 - \mu)^2\right]$$

Now we need to know the prior pdf for the parameter μ. Let the estimated cost/schedule value of the same item at an earlier point "a" along the project be μ'_a, where $\mu_0 = c\mu'_a$. Let μ_a be the actual value of the cost/schedule of the item at this point as reported in the cost performance report (CPR). Now, we can reasonably construct the prior distribution for the parameter μ, by assuming that it is a normal distribution with an expected value of $c\mu_a$, and a standard deviation of $c\sigma_a$, where σ_a is the standard deviation of the item distribution at point "a". Such distribution will thus have

---

2 Zellner, Arnold, An Introduction to Bayesian Inference in Econometrics, John Wiley & Sons, Inc., N. Y. 1971.

the form:

$$p(\mu) = \frac{1}{\sqrt{2\pi}\,\sigma_a c} \exp\left[-\frac{1}{2c^2\sigma_a^2}(\mu - c\mu_a)^2\right]$$

= prior pdf of μ

Combining this prior pdf with the likelihood function, the posterior pdf for the parameter μ becomes:

$$p(\mu \mid \mu_0) \propto p(\mu_0 \mid \mu)\, p(\mu)$$

$$\propto \exp\left[-\frac{1}{2c^2\sigma_a^2}(\mu - c\mu_a)^2 - \frac{1}{2\sigma_0^2}(\mu - \mu_0)^2\right]$$

$$\propto \exp\left[-\frac{1}{2}\left\{\frac{1}{c^2\sigma_a^2}(\mu - c\mu_a)^2 + \frac{1}{\sigma_0^2}(\mu - \mu_0)^2\right\}\right]$$

After some manipulation of the right hand side, we get:

$$p(\mu \mid \mu_0) \propto \exp\left[-\frac{1}{2}\left(\frac{\sigma_0^2 + c^2\sigma_a^2}{c^2 c_0^2 \sigma_a^2}\right)\left\{\mu - \left(\frac{c\mu_a\sigma_0^2 + \mu_0 c^2\sigma_a^2}{\sigma_0^2 + c^2\sigma_a^2}\right)\right\}^2\right]$$

which shows that μ is normally distributed, a posteriori, with mean:

$$E(\mu) = \frac{c\,\mu_a\,\sigma_0^2 + \mu_0 c^2 \sigma_a^2}{\sigma_0^2 + c^2 \sigma_a^2}$$

and variance:

$$V(\mu) = \frac{c^2 \sigma_0^2 \sigma_a^2}{\sigma_0^2 + c^2 \sigma_a^2}$$

Note that the posterior mean and posterior variance could be written in the forms:

$$E(\mu) = \frac{c\,\mu_a\,\sigma_0^2 + \mu_0 c^2 \sigma_a^2}{\sigma_0^2 + c^2 \sigma_a^2} = \frac{c\,\mu_a\,(c\sigma_a)^{-2} + \mu_0 (\sigma_0)^{-2}}{(c\sigma_a)^{-2} + (\sigma_0)^{-2}}$$

and

$$V(\mu) = \frac{c^2 \sigma_0^2 \sigma_a^2}{\sigma_0^2 + c^2 \sigma_a^2} = \frac{1}{(c\sigma_a)^{-2} + (\sigma_0)^{-2}}$$

which shows that the posterior mean is a weighted average of the prior mean $c\,\mu_a$ and the sample mean μ_0, with weights being the reciprocals

of $(c\sigma_a)^2$ and σ_0^2 respectively. If we let $(c\sigma_a)^{-2} = h_a$ and $\sigma_0^{-2} = h_0$ then:

$$E(\mu) = \frac{c\mu_a h_a + \mu_0 h_0}{h_a + h_0}$$

and:

$$V(\mu) = \frac{1}{h_a + h_0}$$

h_a and h_0 being the corresponding precision parameters. Hence the precision parameter associated with the posterior mean is just $\left[V(\mu)\right]^{-1} = h_a + h_0$, the sum of the prior and sample precision parameters.

III. DISCUSSION

This Bayesian approach to the problem of forecasting the cost and schedule of items involved in a Department of Defense contract, is simple and convenient.

Two main assumptions have been made during the course of development of this method: (1) that the cost and schedule at any point along the path of the project are normally distributed. Though not the most realistic, normal distributions are considered a fair approximation and are usually adopted for mathematical convenience; (2) that the prior distribution at point "o" has mean $c\mu_a$ and standard deviation $c\sigma_a$, c being the ratio between the planned cost/schedule at points "o" and "a", respectively. This is tantamount to assuming that the expected value of the cost/schedule at a particular point along the path of the project would relate proportionally with respect to the position of this point on the path. Such assumption is reasonable and logical.

To apply the formulas developed by the present method, it is needed to assign values to the standard deviations σ_a and σ_o. Fair estimates of these two quantities may be obtained by one of two methods: (1) subjective estimates through personnel that are knowledgeable about the particular contract; (2) using the cost/schedule variances reported in earlier cost performance reports (CPR), which are indicative of the dispersion, e.g., assuming they loosely follow a normal distribution, then calculating σ in the usual manner.

The advantages of this Bayesian Scheme are: (1) easy closed formulas are used, which can readily be handled with a desk calculator; (2) the formulas are equally valid at any point along the path of the program, with no extrapolation involved; (3) updating the forecast does not require a new elaborate smoothing or reiterative process, only a reapplication of the formulas by substituting the new data; (4) this is the only plausible method to use in the very early stage of the life of the contract, since then available information is too scanty for any other method to apply meaningfully.

APPENDIX: EXAMPLE

The adjoining table presents an actual contract cost performance report (CPR) of a particular item. The Bayesian Statistical technique will be used to forecast the cost at project completion of this item, projected from these reported data. For cost forecast, we need the following quantities: (1) Budgeted cost of work performed (BCWP), reported in column 8 in CPR; (2) Actual cost of work performed (ACWP) reported in column 9; (3) Budgeted cost at completion reported in column 12. The ACWP is μ_a, and the bugeted cost at completion is μ_0. The quantity c is μ_0/BCWP and the estimated values of σ are, $\sigma_a = 0.1\ \mu_a$ and $\sigma_0 = 0.05\ \mu_0$. Substituting the values of μ_a = 2416.0, μ_0 = 6716.6, σ_a =241.60, σ_0 =335.83, and c = 6716.6/2204.1 = 3.047. Hence:

$$E(\mu) = \frac{c\,\mu_a\sigma_0^2 + \mu_0 c^2 \sigma_a^2}{\sigma_0^2 + c^2\sigma_a^2}$$

$$= \frac{(3.047)(2416.0)(335.83)^2 + (6716.6)(3.047)^2(241.60)^2}{(335.83)^2 + (3.047)^2(241.60)^2}$$

$$= 6827.70 \text{ (in \$1000)}$$

and the statistical variance is

$$V(\mu) = \frac{c^2\sigma_0^2\sigma_a^2}{\sigma_0^2 + c^2\sigma_a^2} = 93353.61$$

$$\sigma = (V)^{1/2} = 305.5 \text{ (in \$1000)}$$

Therefore, based on the CPR reported data at point "a" of the project, the expected cost at completion, point "o", is $6,827,700, with a standard deviation of $305,500.

A similar procedure would be applied to the schedule problem. In this case, μ_a = 2,204.1, viz. the entry in column 8, whereas μ_o will remain the same, i.e., 6716.6 of column 12, and 1/c = 2,286.4/6716.6, where the numerator is the entry in column 7. A value for σ_a will have to be estimated, and calculations will proceed as before.

COST PERFORMANCE REPORT (CPR)

ITEM WBS ELEMENT	CURRENT PERIOD					CUMULATIVE TO DATE					AT COMPLETION		
	BUDGETED COST		ACTUAL COST	VARIANCE		BUDGETED COST		ACTUAL COST	VARIANCE		BUDGETED	LATEST REVISED ESTIMATE	VARIANCE
	WORK SCHED	WORK PERFORMED	WORK PERFORMED	SCHEDULE	COST	WORK SCHED	WORK PERFORMED	WORK PERFORMED	SCHEDULE	COST			
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
Rotor System	314.8	303.4	245.9	(11.4)	57.5	2,286.4	2,204.1	2,416.0	(82.3)	(211.9)	6,716.6	7,209.5	(492.9)

() Indicates Unfavorable variance (behind schedule or over cost)

COMPUTER GRAPHICS APPLIED TO TEACHING OF MATH PRINCIPLES AT USMA.

CPT Arthur G. Bonifas
Department of Mathematics
United States Military Academy
West Point, New York

ABSTRACT. This paper discusses the expanding role of computer graphics in teaching mathematical principles at USMA. Considered are the effects of the computer support facilities and the CAD-E (Computer Aided Design-Engineering) seminar held annually at West Point, New York. A computer graphics demonstration called IFUL which was developed for use in CAD-E, July 1973 by this author is covered in detail. IFUL stands for Integral as a Function of its Upper Limit and shows graphically the limiting process of the derivative,

$$\lim_{h \to 0} \frac{I(x+h) - I(x)}{h},$$

where

$$I(x) = \int_a^x F(t)dt,$$

thus demonstrating graphically that $I'(x) = F(x)$. It also uses this process to convey an idea of the relative accuracy of the rectangle, trapezium and Simpson's methods of approximating the integral. In the program, the accuracy is compared numerically and graphically. Also, a videotape is discussed which demonstrates the concept of the definite integral as area under a curve.

1. INTRODUCTION. The method of teaching mathematics at the United States Military Academy still bears a strong resemblance to the method used in 1802, when the Military Academy first began. The cadets attend their math classes in small sections of from 12 to 16 students with the emphasis placed on daily student participation. They attend class six mornings a week for 75 minutes and receive a grade alsmost daily. The subject matter is treated as a self-study course with the student preparing each lesson the night before and the instructor emphasizing important points and answering questions in class.

Standing in a typical cadet classroom, the only visible training aid in days gone by would have been wall-to-wall blackboards. Here is where the resemblance ends. Today, in addition to hand-powered chalk, we have electron-powered color television sets in every classroom. With this modern day "window" a new dimension has been brought into the traditional West Point mathematics class. The technology of timesharing computer graphics is one of the latest areas in which the Military Academy has attempted to be on the cutting edge of technological development and application to the educational process. Computer graphics allows massive arithmetic calculations to be transformed into simple picture form for quick theoretical analysis by the student and instructor. These abstract principles can actually be observed in action on the screen.

2. VIDEOTAPE 1, (IFUL). The first project to be worked on was a videotape tieing together the geometrical and analytical interpretations of the derivative. We needed a program that would graphically portray taking the derivative of a function. If this function happened to be a definite integral then we would also have the capability to demonstrate that the derivative of a definite integral is the function under the integral or in other words the integral is a function of its upper limit.

2.1 PRINCIPLES. First the videotape illustrates the math principle involved:

Let $F(x) = x^2$, then $G(x) = \frac{x^3}{3}$.

$$I(x) = \int_a^x F(t)dt = \int_a^x t^2 dt = G(x) - G(a)$$

Show that $I'(x) = F(x)$

$$I'(x) = \lim_{h \to 0} \frac{I(x + h) - I(x)}{h}$$

$$= \lim_{h \to 0} \frac{[G(x + h) - G(a)] - [G(x) - G(a)]}{h}$$

$$= \lim_{h \to 0} \frac{G(x + h) - G(a) + G(a) - G(x)}{h}$$

$$= \lim_{h \to 0} \frac{G(x + h) - G(x)}{h} = G'(x) = F(x)$$

Next the cadet sees the geometrical interpretation of the slope of the secant becoming the slope of the tangent line as h in the difference ratio, $\frac{G(x + h) - G(x)}{h}$, goes to zero.

In order to demonstrate this process graphically,

$$\frac{G(x + h) - G(x)}{h}$$

is calculated with h fixed as x assumes successive values. See FIG 1. This gives us curve #1. h is then decreased to a smaller constant value and curve #2 is plotted as x varies. h is decreased five times and five curves are plotted. The last one is seen to converge on the solid curve which is a plot of $F(x) = x^2$. This demonstration is used to generate a discussion of the limiting process of the definition of the derivative in addition to the concept of the integral as a function of its upper limit.

2.2 APPROXIMATIONS. As this program was being written it was noted that with a slight modification we could demonstrate the relative accuracy of integral approximation methods. Instead of using the antiderivative method to calculate I(x), we could use rectangle, trapezoid, or parabola (Simpson's) summation methods. See FIG 2. Thus a visual comparison of the four methods' success in placing the final dotted curve close to the function curve is shown. See FIG 3.

2.3 COMPARISONS. In numerical analysis, however, we're never happy unless we have some numbers to analyze. We needed a numerical measure of the closeness of the final dotted curve to the function curve. If this measure of absolute error is calculated uniformly for each of the four methods of evaluating the integral, we should be able to compare them. ERR= gives this calculation for each method and it is simply the sum of the distances between the final dotted curve and the solid function curve at each value of x. Notice the values in FIG 3. To make the comparison even easier to interpret, these individual absolute error distances are plotted on one set of axes called ERRPLOT. See FIG 4.

These final additions made the program general enough that it could also be used for more advanced study in courses such as the numerical

analysis elective. All the variables in the program can be changed including the function. Thus as summation partitions are increased, the student can see the last dotted curve draw closer to the solid function curve and the absolute error value decrease.

3. VIDEOTAPE 2 (TVTEK). Another videotape was produced to clarify one of the central concepts of integral calculus, the definite integral as area under a curve. The computer plots a function. Then it draws the N number of rectangles representing upper and lower Reimann sums and calculates their respective areas. As N increases, the upper sum decreases and the lower sum increases, until it is obvious to the student that the actual value of the area beneath the curve is bounded by them. See FIGS 5, 6, 7. Thus the role that limiting theory plays in the development of the definite integral as area is clarified.

Computer graphics programs such as these have been found to be extremely helpful in maintaining a high level of academic interest in the lower sections of the basic math courses as well as being a departure point for more advanced sections.

4. CAD-E. We are fortunate at the Academy to have faculty interest and the hardware and software to support an effective computer graphics program. Annually USMA hosts the Computer Aided Design-Engineering Seminar for attendees from Army Material Command Installations around the country. They are familiarized with the graphics system and participate in the use of computer programs written by instructors of the various academic departments. The development of these programs provides a demonstration of the capabilities of graphics to the seminar members and also contributes an effective teaching tool in the form of a completed program to the academic department supplying the instructor. Both of the programs we discussed were written by instructors while working with CAD-E.

5. HARDWARE. The computer graphics hardware available at the Military Academy is extensive and varied. The CPU is a Honeywell H635 with 160K core. The peripherals consist of:

GE Terminet 300 - typewriter termina

Datapoint 3300 - CRT interactive terminal plus TSP 212 - flatbed plotter

Tektronix 4010 - graphics terminal with hard copy unit

Computek 400/15 terminal with a Computek CT50 Graphical Tablet

Imlac PDS-1 - graphics terminal

6. SOFTWARE. In the area of software, we have the USMA-developed GCS, Graphics Compatibility System, which is a FORTRAN-based graphic system of subroutines designed for interactive use on a wide variety of terminals. Terminal compatibility is attained because the user need no longer be concerned about the problems of "tailoring" his program for a given graphical device. User compatibility is attained because individuals who would normally dismiss graphics for their particular application are provided with simple "black box" preset default options, but at the same time more sophisticated users can set these options themselves such as rectangular to polar coordinates, windowing, etc.

7. PRODUCTION. To assist us in getting our computer graphics programs into a usable form, the Instruction Support Division at the Military Academy has a fully operational color television studio, control room and distribution center. Currently we have two graphics videotapes which can be aired in all math classrooms simultaneously at the appropriate course lesson. Also for additional instruction and individual viewing, a copy of each videotape is kept in the Mathematics Library for viewing over a Sony videocassette playback unit.

8. CONCLUSION. So far the investment of time, manpower and materials in producing these computer graphics programs has paid dividends in the classroom. Acceptance by the cadets and instructors has been excellent and plans have been made to create programs for the demonstration of other math topics such as series and projectile motion. See FIGS 8, 9.

9. ACKNOWLEDGEMENT. Videotape commissioning, encouragement, review and approval was conducted by the following individuals in the Department of Mathematics, USMA: COL J. M. Pollin, Professor and Acting Head of Department of Mathematics and COL D. H. Cameron, Associate Professor -

Fourth Class Mathematics. Guidance and assistance was received from MAJ B. R. Arnold, Assistant Professor, Supervisor of Standard Program. Organization, ideas and script came from MAJ S. K. Wasaff, Instructor, Chairman of Instructional Support Committee. Programming and videotape instruction was provided by MAJ G. J. Walk, Assistant Professor.

Background on the Graphical Compatibility System came from the following individuals in the Instruction Support Division, Office of the Dean, USMA: COL W. F. Luebbert, Director, Instruction Support Division. Extensive programming assistance and advice was administered by MAJ H. Gabriel, Chief of the Instructor Group. Professional direction and production of the videotapes was given by Mr. F Baldwin, Audio-Visual Production Officer.

Through the extensive efforts of all the above and many others, computer graphics has become a useful tool available to the mathematics instructor at West Point.

LIMITING CURVES OF THE DERIVATIVE

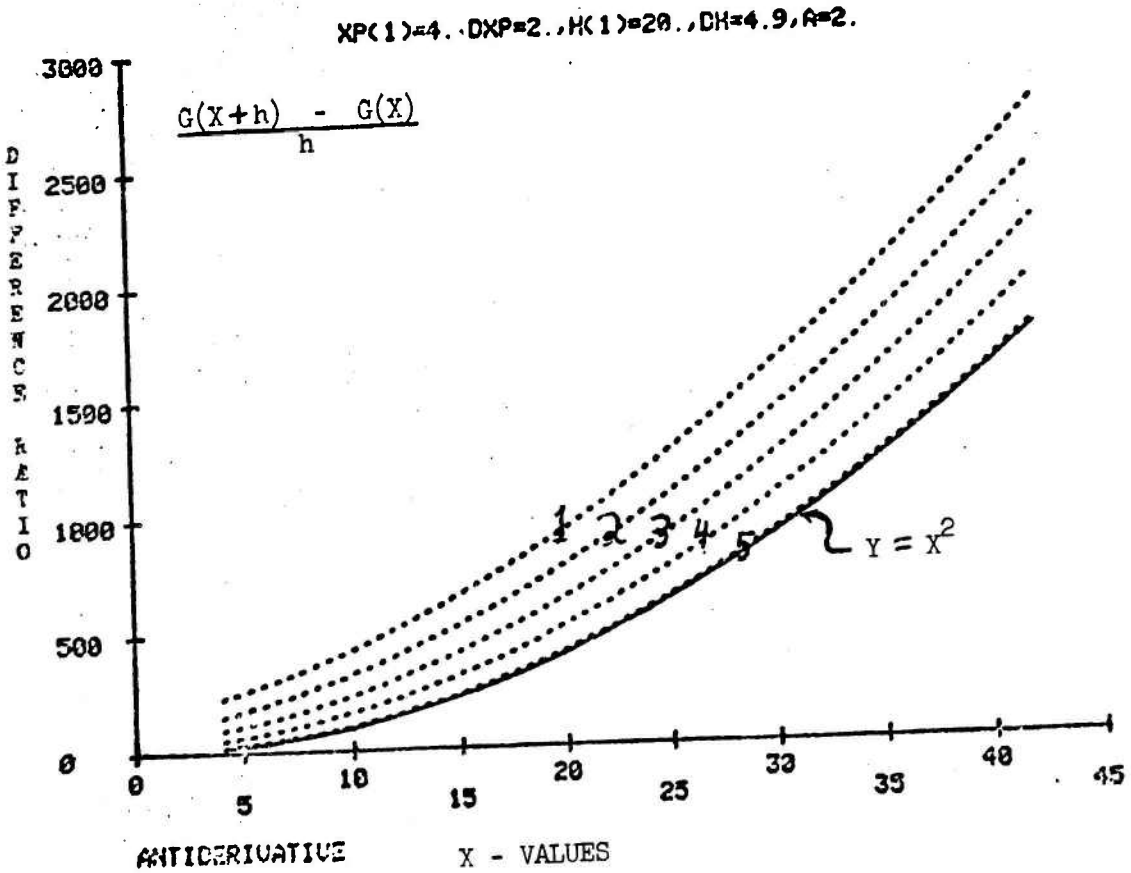


FIG 1

INTEGRAL APPROXIMATION METHODS

(1) Antiderivative method - where G(x) equals the known antiderivative and

$$I(x) = \int_a^x F(t)\,dt \approx G(x) - G(a).$$

(2) Rectangle summation method - where

$$I(x) = \int_a^x F(t)dt \approx \sum_{k=1}^{n} F(x_k)\Delta x$$

(3) Trapezoid summation method - where

$$I(x) = \int_a^x F(t)\,dt \approx \Delta x \sum_{k=1}^{n} F(x_k) + \frac{\Delta x}{2}\,(F(x_o) - F(x_n))$$

(4) Parabola summation method (Simpson's Rule) - where

$$I(x) = \int_a^x F(t)\,dt \approx \tfrac{4}{3}\Delta x(F(x_1) + F(x_3) + \ldots + F(x_{n-1}))$$

$$+ \tfrac{2}{3}\Delta x(F(x_2) + F(x_4) + \ldots + F(x_{n-2})) + \frac{\Delta x}{3}(F(x_o) + F(x_n))$$

FIG 2

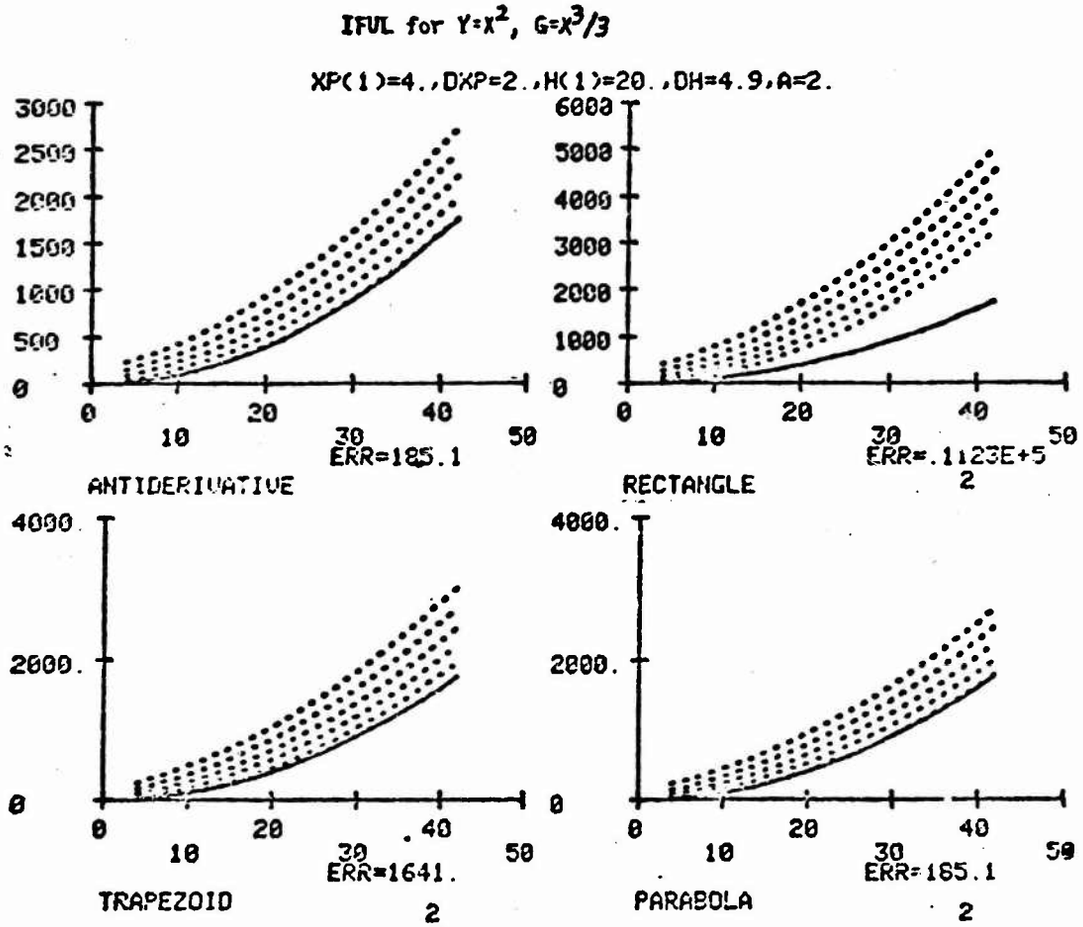

IFUL for Y=X², G=X³/3
XP(1)=4.,DXP=2.,H(1)=20.,DH=4.9,A=2.
3000
2500
2000
1500
1000
500
0
0
10
20
30
40
50
ERR=185.1
ANTIDERIVATIVE
6000
5000
4000
3000
2000
1000
0
ERR=.1123E+5
2
RECTANGLE
4000.
2000.
0
ERR=1641.
2
TRAPEZOID
ERR=185.1
2
PARABOLA

FIG 3.

ERRPLOT

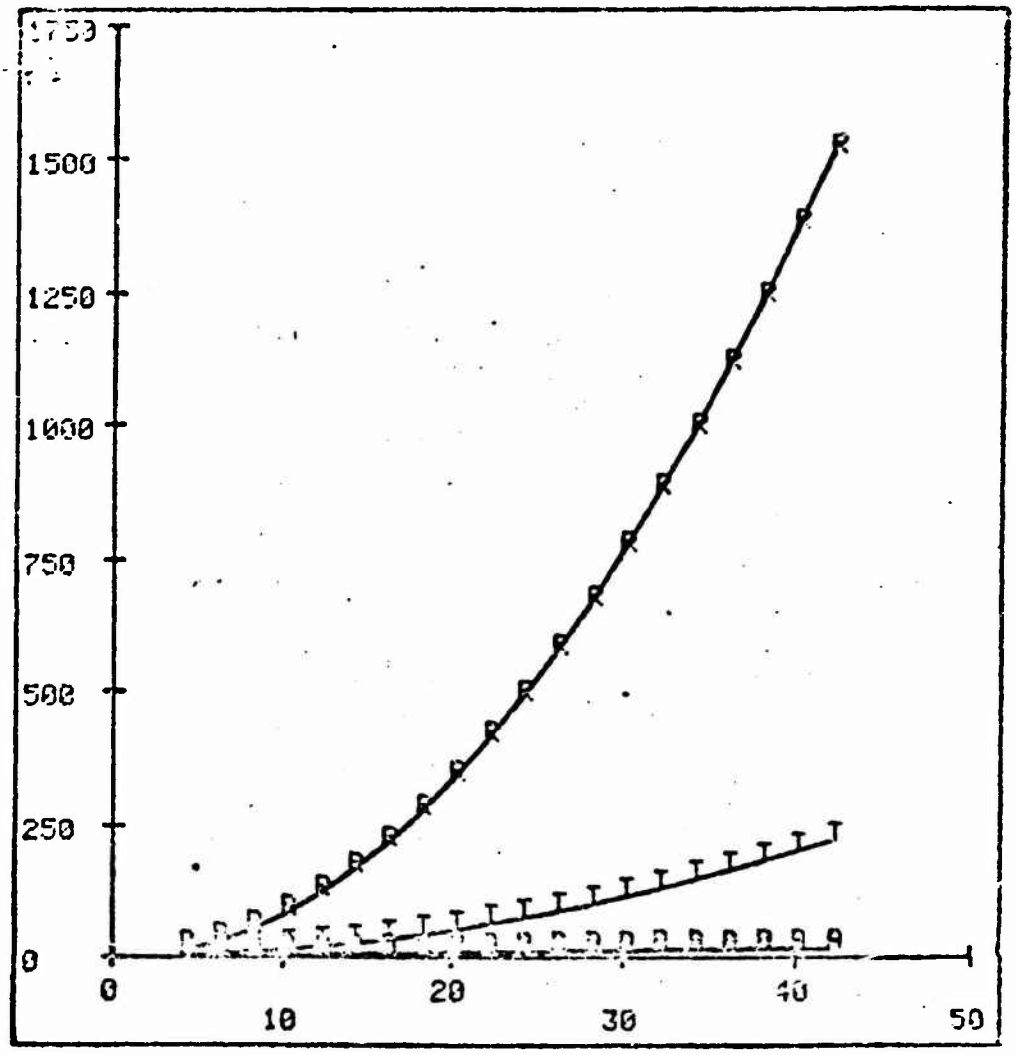


FIG 6

RIEMANN SUMS

$Y=X^2$, $X=0$, $X=3$

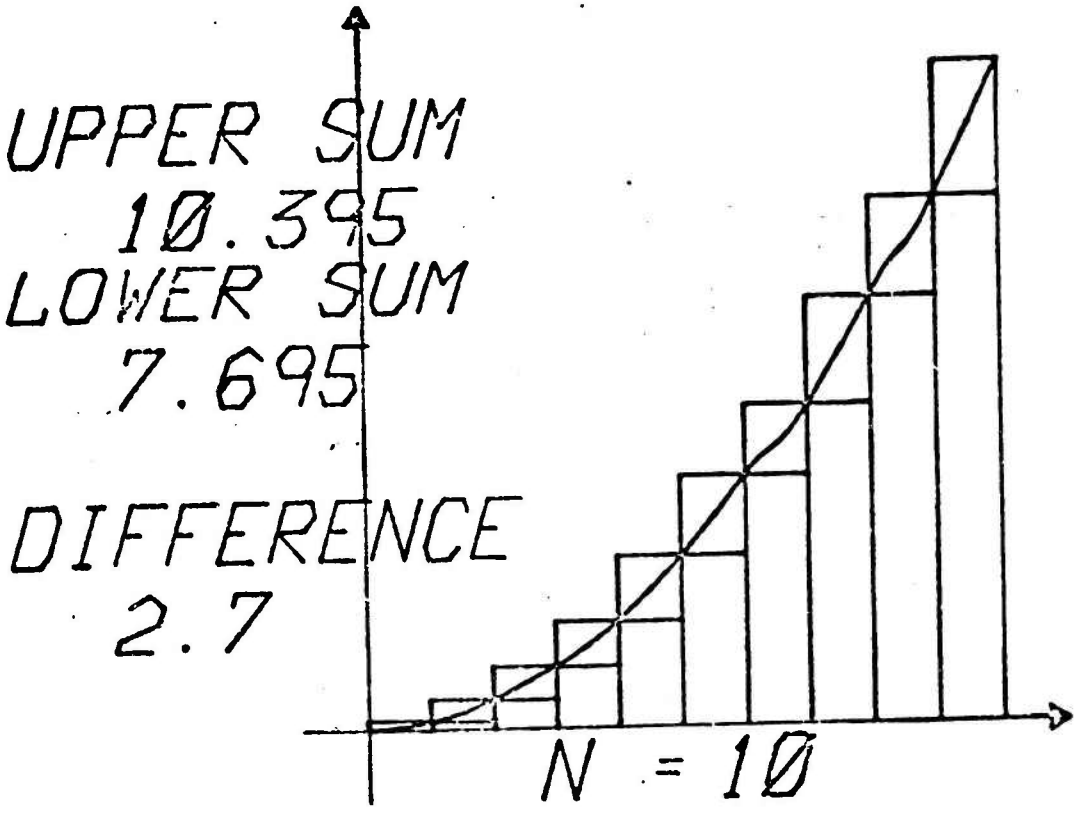


FIG 5

RIEMANN SUMS

$Y=X^2$, $X=1$, $Y=3$

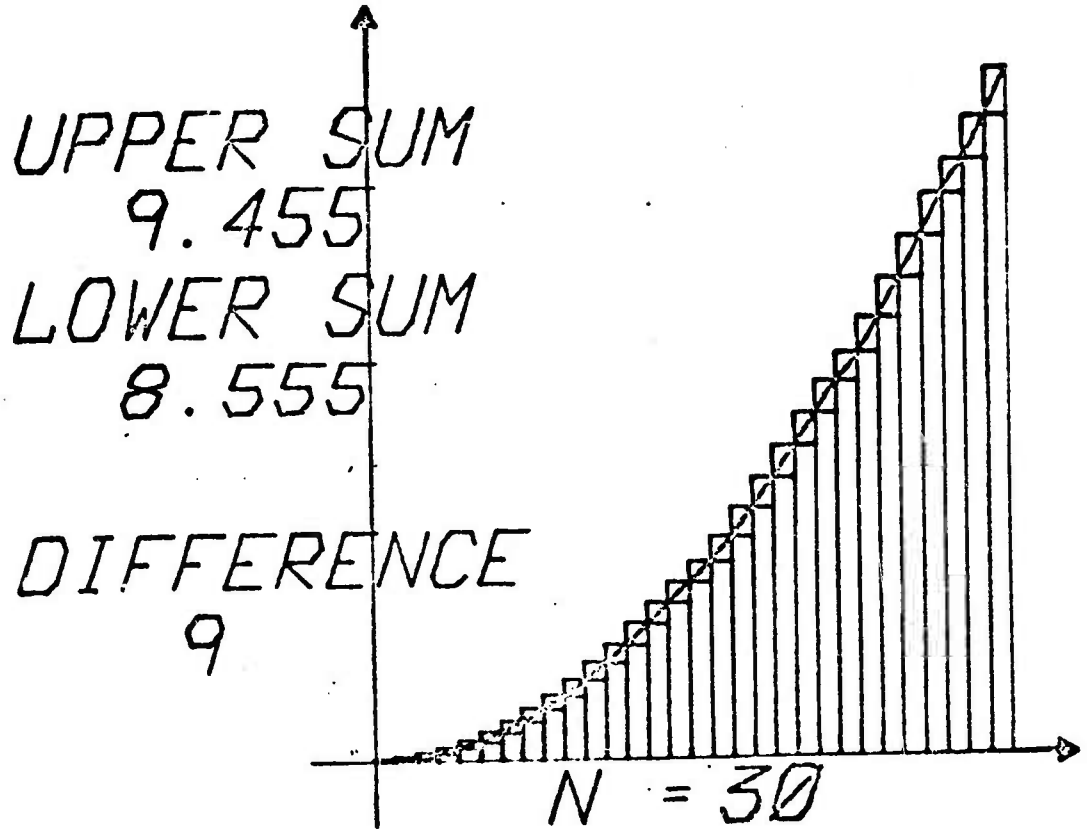


FIG 6

RIEMANN SUMS

$Y=X^2$, $X=0$, $X=3$

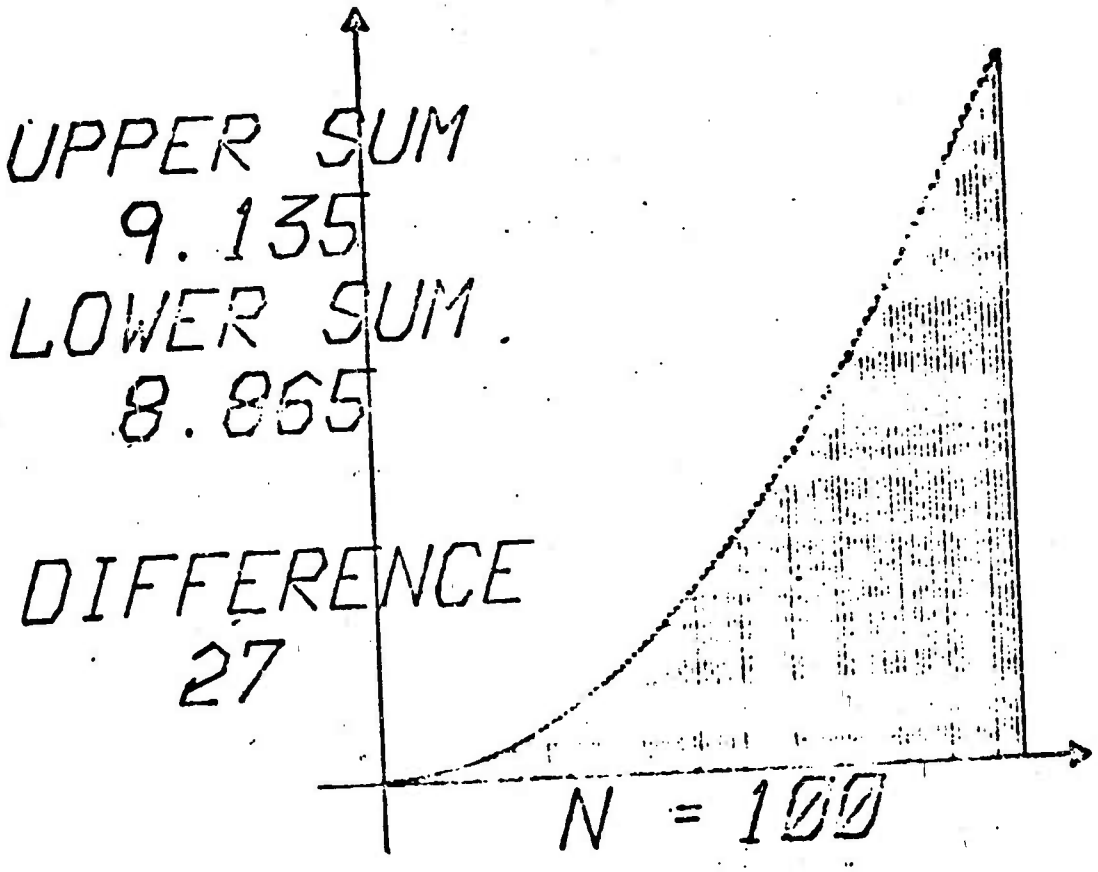


FIG 7

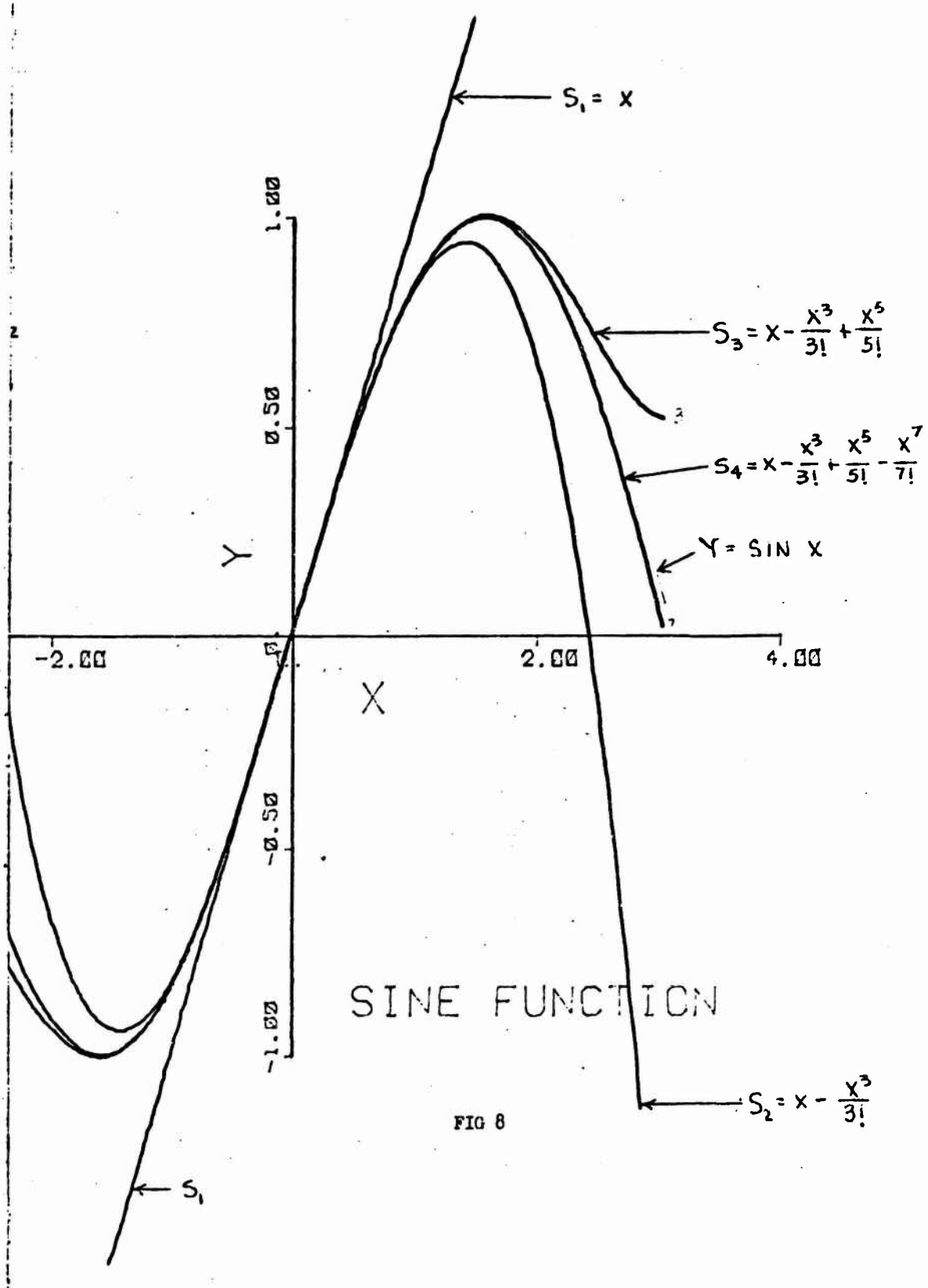

$S_1 = X$
$S_3 = X - \frac{X^3}{3!} + \frac{X^5}{5!}$
$S_4 = X - \frac{X^3}{3!} + \frac{X^5}{5!} - \frac{X^7}{7!}$
$Y = \sin X$
$S_2 = X - \frac{X^3}{3!}$
S_1
1.00
0.50
-0.50
-1.00
-2.00
2.00
4.00
Y
X
SINE FUNCTION

FIG 8

PROJECTILE MOTION

COMPONENT VECTORS

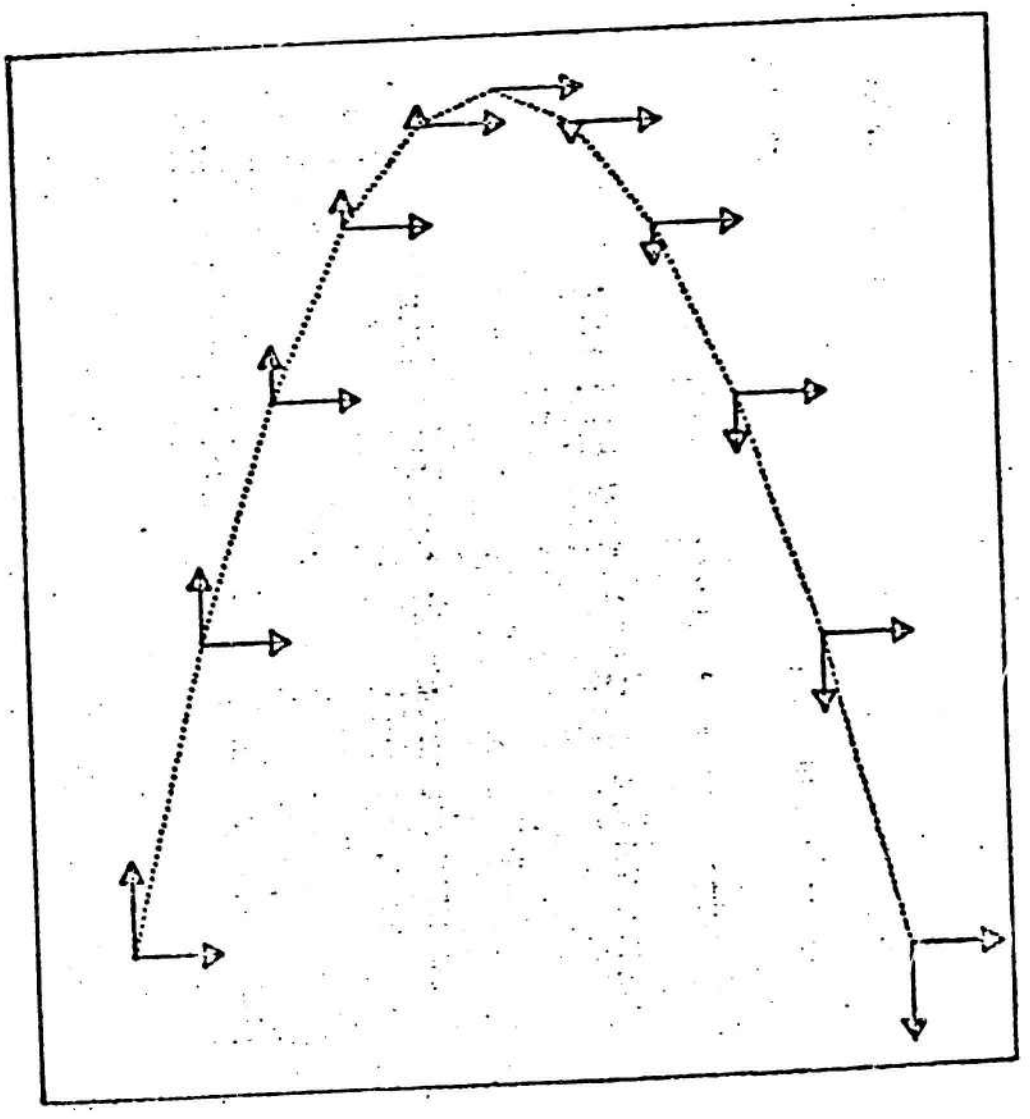


FIG 9

ON THE NUMERICAL CONVERGENCE OF MATRIX EIGENVALUE PROBLEMS DUE TO CONSTRAINT CONDITIONS

Julian J. Wu
Benet Weapons Laboratory
Watervliet Arsenal
Watervliet, New York 12189

ABSTRACT. In the structural analysis of vibrations and stability, the finite element formulations lead to matrix eigenvalue problems. The convergence of the solutions to these problems of linear elasticity has been established for coordinate functions which satisfy all given boundary conditions. This paper is concerned with a numerical convergence study due to approximations of the natural boundary condition. Since it is well known that all boundary conditions can be transformed into natural boundary conditions, this study also includes the effect due to approximations to "geometric boundary conditions."

Recently, the adjoint variational technique has been introduced for the study of dynamic stability due to nonconservative forces. The theoretical basis of convergence has yet to be established. A numerical study of solution convergence of some of these problems is also given in this paper.

1. INTRODUCTION. In obtaining approximate solutions for structural vibrations and stability problems, variational methods are often used. This approach usually involves three steps:

a. The establishment of a variational principle associated with a given boundary value problem — the differential equation and the boundary conditions.

b. The selection of a class of functions with undetermined coefficients (the coordinate functions), from which the approximation to the actual solution is to be picked.

c. Carrying out the extremization procedures so that the approximation is obtained.

The class of coordinate functions are sometimes called admissible functions. Strictly admissible functions satisfy all the boundary conditions, while broadly admissible functions satisfy only the geometric boundary conditions. For the sake of completeness, we recall that the geometric (or imposed) boundary conditions are those imposed on the variations of field variables; and the natural boundary conditions are due to the arbitrariness of the variations as results of the associated variational principles.

The selection of admissible functions is expected to have important effects on the convergence of the approximations and this study intends to evaluate such effects in numerical terms and in conjunction with the finite element analysis.

The finite element method can be viewed as a Rayleigh-Ritz type of approximation using piecewise analytic coordinate functions. It is well established that its procedure will converge for problems of linear elasticity [1]. In using the Ritz method with analytic functions over the entire region of the problem, it is also known that broadly admissible functions will converge [2]. Then it is conceivable that the broadly admissible piecewise analytic functions used in the finite element analysis will also lead to convergence.

Furthermore, it has been shown in the literature that, through a limiting process, geometric boundary conditions can be transformed into natural boundary conditions [3]. Hence, in doing so, the class of admissible functions can be further broadened to include functions which do not satisfy any boundary conditions — we shall refer to this class of functions as the unconstrained field variables.

Thus, in more specific terms, the purpose of this investigation is to study the effects on the rate of convergence when the different classes of admissible functions are used.

The importance of this information can be seen in two aspects.

First, the rate of convergence has direct bearing on the computer time used. The best choice of admissible functions thus implies the optimal use of computers.

Secondly, it is generally laborious to find strictly, or even broadly admissible functions. This extra work must be justified by the economy of computer time saved if there should be any savings at all.

2. STATEMENT OF THE PROBLEM. We shall consider a cantilevered column subject to a concentrated force as shown in Figure 1. The differential equation is

$$EIu^{IV} + P\,u'' + \rho A\,\ddot{u} = 0 \qquad (1)$$

where

A, I = area, second moment of the cross section
ρ, E = mass per unit length, Young's modulus of the material
P = the concentrated force applied at the free end
u = lateral deflection of the column.

A prime (') or a roman numerial denotes a differentiation with respect to the spatial coordinates x, and a dot ($\cdot$) denotes a differentiation with respect to time t.

The boundary conditions at the fixed end are

$$u(0) = u'(0) = 0. \tag{2}$$

At the free end, two different sets of boundary conditions will be considered:

$$\begin{aligned} EI\, u'''(1) + P\, u(1) &= 0 \\ u''(1) &= 0 \end{aligned} \tag{3}$$

and

$$u'''(1) = u''(1) = 0. \tag{4}$$

Equations (3) pertain to the case of a force with constant direction parallel to the undeformed axis of the column, and equation (4), to that with a follower force which remains tangential to the column at the free end. These two sets of boundary conditions result in two quite different boundary value problems — one associated with a self adjoint system, the other, with a nonself-adjoint system.

Using dimensionless variables, it is easy to see that the two problems can be written as the following:

Problem I (Self-adjoint)

$$\text{D.E.} \quad u^{IV} + Q\, u'' - \lambda u = 0 \tag{5}$$

$$\text{B.C.} \quad u(0) = u'(0) = 0 \quad \text{(imposed)} \tag{6}$$

$$\text{B.C.} \quad u''(1) = u'''(1) + Q\, u(1) = 0 \quad \text{(natural)} \tag{7}$$

Problem II (Nonself-adjoint)

$$\text{D.E.} \quad u^{IV} + Q\, u'' - \lambda u = 0 \tag{8}$$

$$\text{B.C.} \quad u(0) = u'(0) = 0 \quad \text{(imposed)} \tag{9}$$

$$\text{B.C.} \quad u''(1) = u'''(1) = 0 \quad \text{(natural)} \tag{10}$$

where $Q = \dfrac{P\ell^2}{EI}$ is the dimensionless load parameter. Since we are seeking solutions periodic in time, $\ddot{u}$ is replaced by $-\lambda u$ without loss of generality.

The imposed boundary conditions of equations (6) or (9) can be transformed into natural boundary conditions as the following:

$$\lim_{k_1 \to \infty} [u'''(0) + k_1 u(0)] = 0$$

$$\lim_{k_2 \to \infty} [u''(0) + k_2 u'(0)] = 0 . \tag{11}$$

We shall write down the variational principles associated with the above mentioned boundary value problems. They are the basis of our finite element analysis.

In association with equations (5), (6) and (7),

$$\begin{aligned} \delta J_1 &= 0 \\ J_1 &= \frac{1}{2} \int_0^1 [(u'')^2 - Q(u')^2 - \lambda u^2]dx. \end{aligned} \tag{12}$$

With equations (5), (11) and (7), we have

$$\begin{aligned} \delta J_2 &= 0 \\ J_2 &= \frac{1}{2} \int_0^1 [u'')^2 - Q(u')^2 - \lambda u^2]dx \\ &\quad + k_1 [u(0)]^2 + k_2[u'(0)]^2. \end{aligned} \tag{13}$$

With equations (8), (9) and (10), we have

$$\begin{aligned} \delta J_3 &= 0 \\ J_3 &= \int_0^1 (u''v'' - Q u'v' - \lambda uv)dx \\ &\quad + Qu'(1)v(1). \end{aligned} \tag{14}$$

And with equations (8), (11) and (10), we have

$$\begin{aligned} \delta J_4 &= 0 \\ J_4 &= \int_0^1 (u''v'' - Qu'v' - \lambda uv)dx \\ &\quad + Qu'(1)v(1) + k_1 u(0)v(0) + k_2 u'(0)v(0) \end{aligned} \tag{15}$$

where v is the adjoint field variable.

Using the finite element method, this investigation intends to compare the rate of numerical convergence among the following cases:

A. When the coordinate functions satisfy all the boundary conditions (equations (6), (7) and (12) for Problem I and equations (9), (10) and (14) for Problem II).

B. When the coordinate functions satisfy only the geometric (imposed) boundary conditions (equations (7) and (12) for Problem I and equations (10) and (14) for Problem II).

C. When coordinate functions do not satisfy any boundary conditions (equation (13) for Problem I and equation (15) for Problem II).

3. FINITE ELEMENT FORMULATIONS. For the finite element analysis, the column in question is divided into several segments (elements) as shown in Figure 1(D). Only an outline of this formulation will be given here to introduce some terminology. The details have been provided in a previous paper [4].

Let us consider Case A of Problem I described in the previous section: a self-adjoint problem with all boundary conditions satisfied. In discrete system, the variational principle takes the following form:

$$\delta J_1 = 0 \tag{16}$$

$$J_1 = \sum_{i=1}^{L} J_1^{(i)} \tag{17}$$

and

$$J_1^{(i)} = \frac{1}{2}\int_0^1 [L^3(u^{(i)''})^2 - QL(u^{(i)'})^2 - \frac{\omega^2}{L}(u^{(i)})^2]d\xi, \tag{18}$$

where L is the number of elements,

$$\xi = L\left(x - \frac{i-1}{L}\right) \tag{19}$$

$$u^{(i)} = a^T(\xi)\, U^{(i)} \tag{20}$$

$$\begin{aligned} a^T(\xi) &= \{a_1(\xi) \quad a_2(\xi) \quad a_3(\xi) \quad a_4(\xi)\} \\ &= \{1-3\xi^2+2\xi^3 \quad \xi-2\xi^2+\xi^3 \quad 3\xi^2-2\xi^3 \quad -\xi^2+\xi^3\} \end{aligned} \tag{21}$$

$$U^{(i)T} = \{U_1^{(i)} \quad U_2^{(i)} \quad U_3^{(i)} \quad U_4^{(i)}\}, \tag{22}$$

and a superscript T denotes the transpose of a matrix.

Introducing the following matrices:

$$\underset{\sim}{A} = \int_0^1 a(\xi)\, a^T(\xi) d\xi \tag{23}$$

$$\underset{\sim}{B} = \int_0^1 a'(\xi)\, a'^T(\xi) d\xi \tag{24}$$

$$\underset{\sim}{C} = \int_0^1 a''(\xi)\, a''^T(\xi) d\xi. \tag{25}$$

Equation (18) can be written as,

$$J_1^{(i)} = \frac{1}{2} U^{(i)T} \{L^3 \underset{\sim}{C} - QL \underset{\sim}{B} + \frac{\omega^2}{L} \underset{\sim}{A}\} U^{(i)}$$

$$= \frac{1}{2} U^{(i)T} k\, U^{(i)} \tag{26}$$

where

$$\underset{\sim}{k} = L^3 \underset{\sim}{C} - QL \underset{\sim}{B} + \frac{\omega^2}{L} \underset{\sim}{A} \tag{27}$$

is the "element stiffness matrix".

Applying equation (16) and using the continuity requirement,

$$U_3^{(i-1)} = U_1^{(i)}$$

$$U_4^{(i-1)} = U_2^{(i)} \qquad i = 1,2,\ldots,L. \tag{28}$$

We can arrive at the following equation

$$\delta U^T K U = 0 \tag{29}$$

where

$$U^T = \{U_3^{(1)}\ U_4^{(1)}\ U_3^{(2)}\ U_4^{(2)} \ \ldots\ldots\ U_3^{(L-1)}\ U_4^{(L-1)}\} \tag{30}$$

and K is obtained by assembling $\underset{\sim}{k}$ properly.

It is important to note here that in obtaining equation (29), boundary conditions, (equations (6) and (7)), also have been applied, since the coordinate functions are required to satisfy constraint conditions. These conditions can be written as:

$$
\begin{aligned}
U_1^{(1)} &= 0 \\
U_2^{(1)} &= 0 \\
3U_1^{(L)} + U_2^{(L)} - 3U_3^{(L)} + 2U_4^{(L)} &= 0 \\
2U_1^{(L)} + U_2^{(L)} - (2 - \tfrac{Q}{6})\, U_3^{(L)} + U_4^{(L)} &= 0.
\end{aligned}
\tag{31}
$$

The process of using equations (31) in obtaining equation (29) can be quite complicated, especially for the case of nonself-adjoint problems. This constitutes one advantage when the unconstrained field variables are used with the proper variational principles.

Since δU in equation (29) is now arbitrary, we have the matrix eigenvalue equation

$$K(\omega^2)U = 0. \tag{32}$$

Equation (32) will be used for the eigenvalue calculations. Similar procedures will also lead to equations as equation (32) for other cases.

4. CONVERGENCE DATA AND DISCUSSIONS. The data obtained in this study are from a computer system IBM 360, Model 44 and the associated Operating System.

Problem I. Free Vibrations of a Cantilevered Column (Figure 1(A)) - A Self-Adjoint Problem.

In Tables 1 through 3, approximate eigenvalues are given for the first six modes of vibrations and with the number of elements used increasing from two (2) to eight (8). The percentage of error compared with the exact values are given in parentheses. In Table 1, results are for the case when all constraint conditions are satisfied. Tables 2 and 3 are for the cases when the geometric conditions only are satisfied, and, when none of the boundary conditions are satisfied, respectively.

A close examination of these results shows clearly that for the free vibrations of a cantilevered column, the frequencies can most efficiently be obtained by using the unconstrained field variables in conjunction with a properly chosen variational principle. This means: although the coordinate functions need not satisfy any constraint conditions, the "energy" terms contributed by these constraints must be included in the variational statement.

As indicated in the previous section, the selection of a class of coordinate functions which satisfy the natural boundary conditions can be a complicated task. This additional labor must be justified by savings in computer time. However, results in Tables 1 and 2 show that this requirement actually impede the rate of convergence of the eigenvalue calculations. The effect due to the requirements on the geometric boundary conditions is not so pronounced as can be seen in Tables 2 and 3. These results have demonstrated, nevertheless, that such requirements are unnecessary to obtain better convergent solutions.

Problem II. A Cantilevered Column Under a Concentrated Load With Fixed Direction (Euler's Column, Figure 1(B)) - A Self-Adjoint Problem.

Here the critical load is obtained for Euler's problem using the dynamic method. In Figure 2, results are shown for the approximations using three (3) elements. Again, better approximation to the exact value of the critical load ($Q/\pi^2 = 0.250$) is given by using the unconstrained coordinate functions ($Q/\pi^2 = 0.251$); the error is less than one percent. When the coordinate functions satisfy all the constraint conditions, three elements approximation gives a solution ($Q/\pi^2 = 0.265$) which is about six percent in error. Same conclusion holds when five elements are used. This is shown in Figure 3. The solid curve in Figure 3 coincides quite well with the exact solution curve.

Problem III. A Cantilevered Column Under a Concentrated Follower Force at the Free End (Beck's Problem, Figure 1(C)) - A Nonself-Adjoint Problem.

Since this is a nonself-adjoint problem, coordinate functions for the adjoint problem must be included in the formulations. Consequently, the task to satisfy the constraint conditions is twice more complicated. Once again as shown in Figures 4 and 5, better rate of convergence is achieved by the use of the unconstrained coordinate functions.

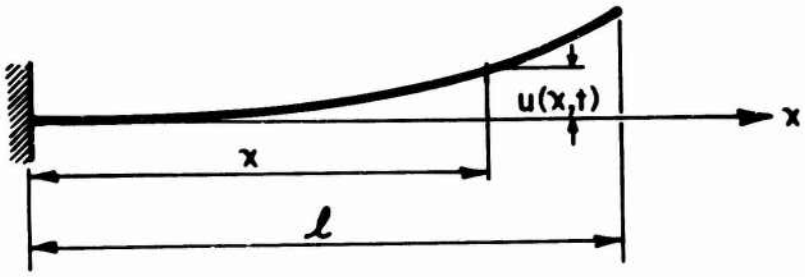
REFERENCES

1. P. TONG and T.H.H. Pian, The Convergence of Finite Element Method in Solving Linear Elastic Problems. Int. Journal Solids, Structures, 3 pp. 865-879 (1967).

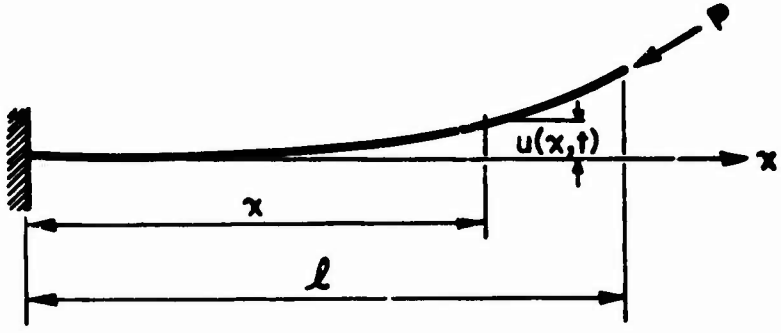
2. K. N. TONG, Theory of Mechanical Vibration, John Wiley and Sons, New York, 1960, pp. 282-285.

3. R. COURANT and D. HILBERT, Methods of Mathematical Physics, McGraw-Hill, New York, 1953, p. 210.

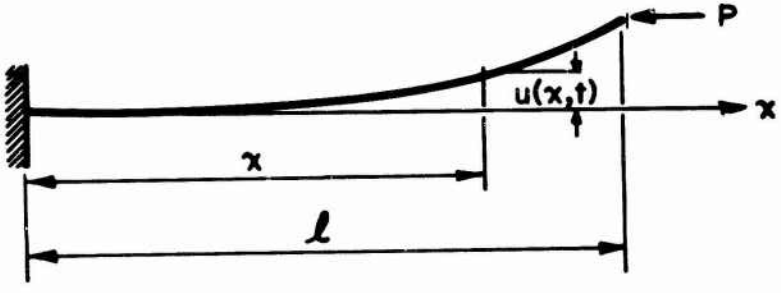
4. J. J. WU, Column Instability Under Nonconservative Forces, With Internal and External Damping — Finite Element Using Adjoint Variational Principles, Development in Mechanics, Volume 7, pp. 501-514 (1973).



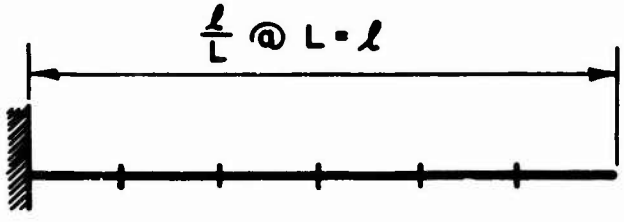
(A) PROBLEM ONE: FREE VIBRATIONS



(C) PROBLEM THREE: BECK'S COLUMN



(B) PROBLEM TWO: EULER'S COLUMN



(D) FINITE ELEMENT MODEL

FIGURE 1:

PROBLEM CONFIGURATIONS AND THE FINITE ELEMENT IDEALIZATION

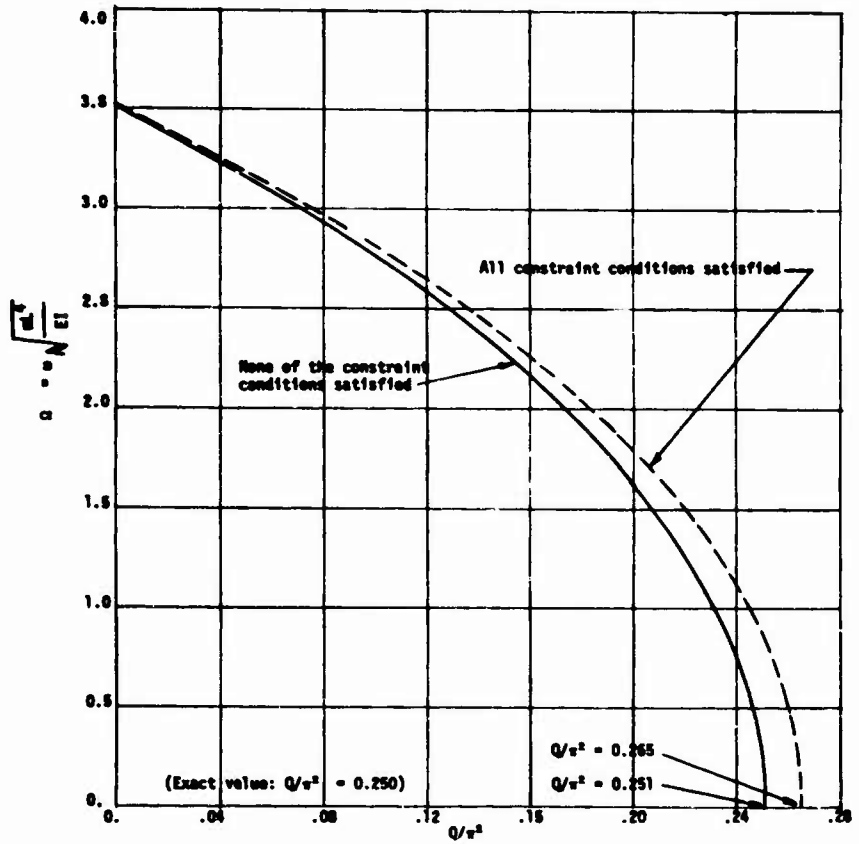


FIGURE 2: BULER'S PROBLEM: THREE(3) ELEMENTS USED

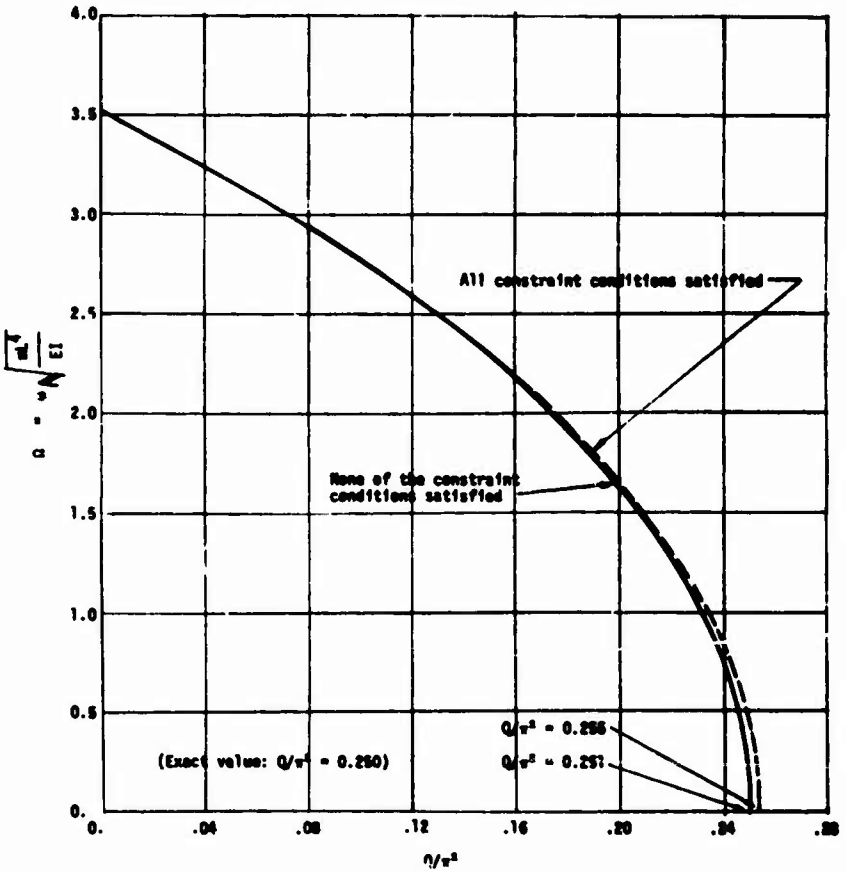


FIGURE 3: BULER'S PROBLEM: FIVE(5) ELEMENTS USED

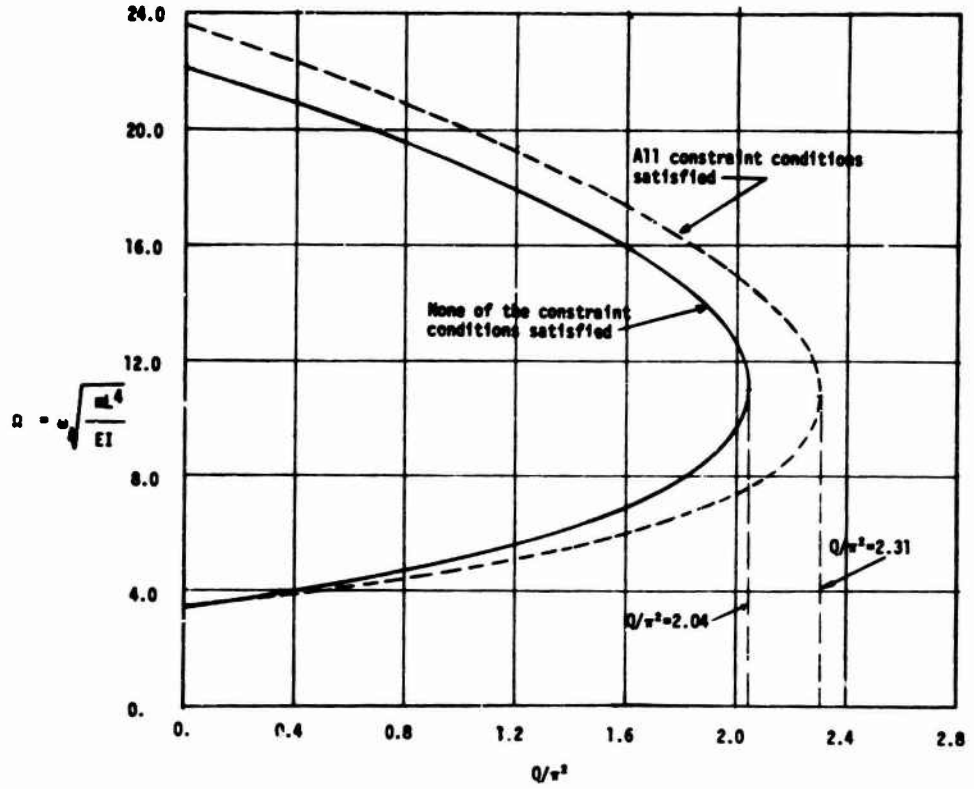


FIGURE 4: BECK'S PROBLEM: THREE(3) ELEMENTS USED

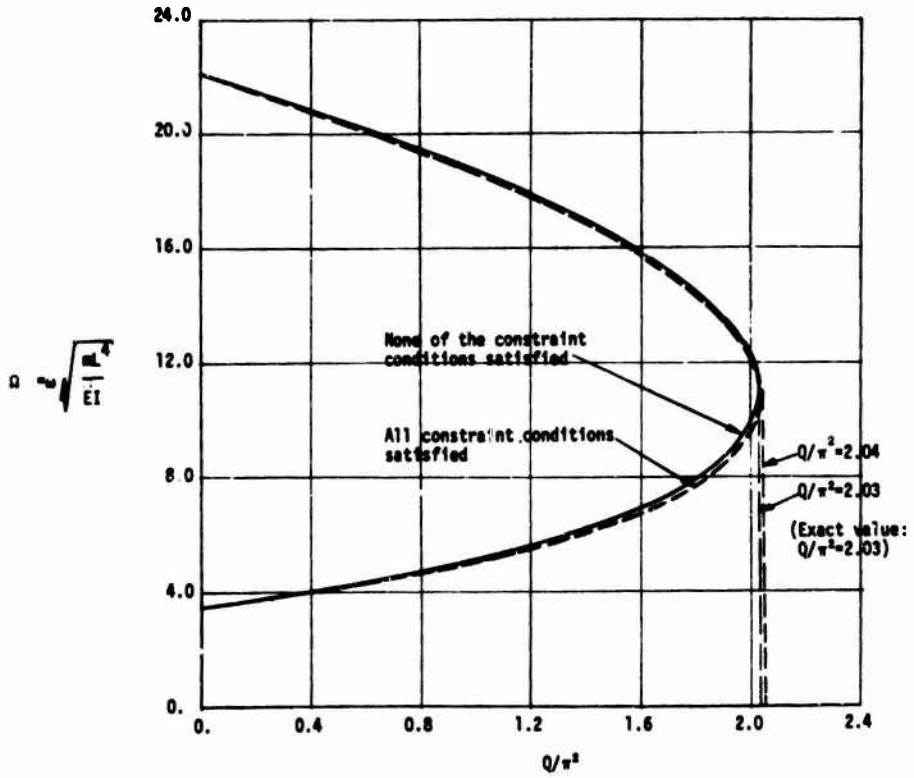


FIGURE 5: BECK'S PROBLEM: FIVE(5) ELEMENTS USED

TABLE 1

APPROXIMATIONS TO EIGENVALUES (PERCENTAGE OF ERROR)

CASE 1, ALL CONSTRAINT CONDITIONS SATISFIED

NO. OF ELEMENTS / MODE NO.	2	3	4	5	6	7	8	EXACT VALUES
1	3.6077 (2.608)	3.5297 (0.330)	3.5195 (0.100)	3.5172 (0.054)	3.5165 (0.014)	3.5163 (0.009)	3.5161 (0.005)	3.5160
2	29.575 (34.224)	23.599 (7.105)	22.528 (2.242)	22.227 (0.876)	22.122 (0.399)	22.078 (0.200)	22.058 (0.109)	22.054
3		77.817 (26.126)	67.792 (9.872)	64.380 (4.342)	63.023 (2.143)	62.409 (1.147)	62.105 (0.655)	61.701
4		215.58 (78.298)	145.32 (20.019)	133.43 (10.355)	127.57 (5.508)	124.71 (3.135)	123.20 (1.894)	120.91
5			293.68 (46.950)	233.03 (16.602)	220.28 (10.223)	212.09 (6.125)	207.55 (3.853)	199.85
6			547.88 (83.508)	402.06 (34.666)	340.90 (14.381)	328.06 (9.881)	317.72 (6.417)	298.56

TABLE 2

APPROXIMATIONS TO EIGENVALUES (PERCENTAGE OF ERROR)

CASE 2, ONLY GEOMETRIC CONDITIONS SATISFIED

NO. OF ELEMENTS / MODE NO.	2	3	4	5	6	7	8	EXACT VALUES
1	3.5177 (0.048)	3.5164 (0.011)	3.5161 (0.003)	3.5161 (0.003)	3.5160 (0.000)	3.5160 (0.000)	3.5160 (0.000)	3.5160
2	22.221 (0.849)	22.107 (0.331)	22.060 (0.118)	22.046 (0.054)	22.040 (0.027)	22.037 (0.014)	22.036 (0.009)	22.054
3	75.157 (21.808)	62.466 (1.240)	62.175 (0.768)	61.919 (0.353)	61.810 (0.177)	61.760 (0.096)	61.735 (0.055)	61.701
4	218.13 (80.407)	140.67 (16.343)	122.66 (1.447)	122.32 (1.166)	121.68 (0.637)	121.35 (0.364)	121.17 (0.215)	120.91
5		264.74 (32.469)	228.14 (14.156)	205.02 (1.586)	202.86 (1.506)	201.71 (0.931)	201.02 (0.585)	199.85
6		527.80 (76.782)	386.39 (22.719)	337.27 (12.966)	303.53 (1.665)	303.04 (1.768)	302.11 (1.189)	298.56

TABLE 3

APPROXIMATIONS TO EIGENVALUES (PERCENTAGE OF ERROR)

CASE 3, NONE OF THE CONSTRAINT CONDITIONS SATISFIED

NO. OF ELEMENTS / MODE NO.	2	3	4	5	6	7	8	EXACT VALUES
1	3.5177 (0.048)	3.5164 (0.011)	3.5161 (0.005)	3.5160 (0.000)	3.5160 (0.000)	3.5160 (0.000)	3.5160 (0.000)	3.5160
2	22.220 (0.844)	22.106 (0.327)	22.059 (0.113)	22.044 (0.045)	22.039 (0.023)	22.037 (0.014)	22.036 (0.009)	22.054
3	75.146 (21.791)	62.458 (1.227)	62.167 (0.755)	61.911 (0.340)	61.802 (0.184)	61.759 (0.094)	61.734 (0.053)	61.701
4	218.09 (80.574)	140.63 (16.310)	122.63 (1.423)	122.29 (1.141)	121.65 (0.612)	121.35 (0.363)	121.17 (0.213)	120.91
5	2623.2 (1212.6)	264.62 (32.408)	228.09 (14.106)	202.95 (1.551)	202.78 (1.486)	201.70 (0.926)	201.01 (0.580)	199.85
6	79800. (26631.)	527.70 (76.798)	386.14 (22.695)	337.08 (12.902)	303.37 (1.611)	303.02 (1.782)	302.08 (1.182)	298.56

AN IMBEDDING METHOD

FOR

NONLINEAR MATRIX EIGENVALUE PROBLEMS

OF STABILITY AND VIBRATION

R. E. Kalaba
Department of Economics and Biomedical Engineering
University of Southern California
Los Angeles, California 90007

M. R. Scott*
Sandia Laboratories
Albuquerque, New Mexico 87115

E. Zagustin
Civil Engineering Department
California State University
Long Beach, California 90840

Summary

In many engineering problems in the theory of stability and vibrations we must find the roots of an equation of the form $\det B(\lambda) = 0$ where $B(\lambda)$ is a nonlinear matrix of λ. In this paper we show how this problem may be reduced to integrating a system of ordinary differential equations subject to initial conditions. The method covers the case of complex roots, and, when specialized to the case of $B(\lambda)$ being linear in λ provides an approach to the usual eigenvalue problem.

---

* This work was supported by the U.S. Atomic Energy Commission

1. Reduction of Nonlinear Eigenvalue Problem to an Initial Value Problem.

Consider the square matrix $B(\lambda)$, where λ is a certain parameter which can enter in a linear or nonlinear form. This type of matrix occurs in the determination of eigenfrequencies (either in buckling or in vibration problems).

In order to find the eigenvalues λ, the following condition must be satisfied

$$\Delta = 0 \qquad \Delta = \det B(\lambda). \tag{1}$$

In order to reduce this problem to an initial value problem, i.e., to a system of differential equations with given initial conditions (which is easily solved computationally), let us introduce the matrix M, which is the adjoint of the matrix B and whose elements are the cofactors of the i^{th} and j^{th} element of B, i.e.

$$M = \text{adj } B(\lambda) \tag{2}$$

where

$$\text{adj } B = (b_{ji}) \tag{3}$$

and $B_{i,j}$ is the cofactor of the i, j^{th} element.
Then the inverse of the matrix B is given

$$B^{-1} = \frac{\text{adj } B}{\det B}, \tag{4}$$

$$B^{-1} = \frac{M}{\det B}. \tag{5}$$

Now, premultiplying both sides of (4) by the matrix B we get:

$$BB^{-1} = B \frac{\text{adj } B}{\det B} . \tag{6}$$

Recalling that

$$BB^{-1} = I, \tag{7}$$

where I is a unit matrix, and postmultiplying both sides of (6) by the det B, we have

$$I \det B = B \text{ adj } B, \tag{8}$$

by postmultiplying both sides of (4) by B.det (B) we have

$$I \det B = (\text{adj } B)\, B. \tag{9}$$

In order to obtain a Cauchy system, let us differentiate both sides of equation (8) with respect to the parameter λ , which yields

$$B_\lambda \text{ adj } B + B(\text{adj } B)_\lambda = I\, (\det B)_\lambda . \tag{10}$$

By premultiplying both sides of equation (10) by adj B we get

$$(\text{adj } B)\, B_\lambda \text{adj } B + (\text{adj } B)\, B\, (\text{adj } B)_\lambda = (\text{adj } B)_\lambda I\, (\det B)_\lambda . \tag{11}$$

By making use of equation (9), i.e.,

$$(\text{adj } B)\, B = (\det B)\, I, \tag{12}$$

in the second term of equation (11) we obtain

$$(\text{adj } B)\, B_\lambda (\text{adj } B) + (\det B)\, (\text{adj } B)_\lambda = (\text{adj } B)\, (\det B)_\lambda \tag{13}$$

Since det B is a scalar, from equation (13) we find

$$(\text{adj } B)_\lambda = \frac{(\text{adj } B)\, (\det B)_\lambda - (\text{adj } B)\, B_\lambda\, (\text{adj } B)}{\det B} . \tag{14}$$

Let us call b_{ij} the element of the i-th row and j-th column of the matrix B. Then differentiating the det B with respect to λ we obtain

$$(\det B)_\lambda = \sum_{i,\,j=1}^{N} \frac{\partial(\det B)}{\partial b_{ij}} \frac{db_{ij}}{d\lambda} \tag{15}$$

But

$$\frac{\partial}{\partial b_{ij}} (\det B) = B_{ij}, \tag{16}$$

where B_{ij} is the cofactor of the element in the i^{th} row and j^{th} column.

Now, substitution of equation (16) into (15) yields

$$(\det B)_\lambda = \sum_{i,j=1}^{N} B_{ij} \frac{db_{ij}}{d\lambda} . \tag{17}$$

Let us now evaluate the product of

$$(\text{adj } B)\, B_\lambda .$$

Let us denote by b^1_{ij} the elements of B_λ . Then we have

$$(\text{adj } B)\, B_\lambda = \begin{pmatrix} B_{11} & B_{21} & \cdots & B_{n1} \\ B_{12} & B_{22} & \cdots & B_{n2} \\ \vdots & & & \\ B_{1n} & B_{2n} & \cdots & B_{nn} \end{pmatrix} \begin{pmatrix} b^1_{11} & b^1_{12} & \cdots & b^1_{1n} \\ b^1_{21} & b^1_{22} & \cdots & b^1_{2n} \\ \vdots & & & \\ b^1_{n1} & b^1_{n2} & \cdots & b_{nn} \end{pmatrix} \tag{18}$$

The terms on the principal diagonal of this product will be

$$(\text{adj } B)\, B_\lambda = \begin{pmatrix} (B_{11}b^1_{11} + B_{21}b^1_{21} + \cdots B_{n1}b^1_{n1}) & & & \\ & (B_{12}b^1_{12} + B_{22}b^1_{22} + \cdots B_{n2}b^1_{n2}) & & \\ & & \ddots & \\ & & & (B_{1n}b^1_{1n} + B_{2n}b^1_{2n} + \cdots B_{nn}b^1_{nn}) \end{pmatrix}$$

The trace of the product (adj B).B_λ is by definition the sum of all the terms on the principal diagonal, i.e.

$$\text{trace}\left[(\text{adj } B)\, B\lambda\right] = (B_{11}b^1_{11} + B_{21}b^1_{21} + \ldots\ldots B_{n1}b^1_{n1})$$
$$+ (B_{12}b^1_{21} + B_{22}b^1_{22} + \ldots\ldots B_{n2}b^1_{n2})$$
$$\vdots$$
$$+ (B_{1n}b^1_{1n} + B_{2n}b^1_{2n} + \ldots\ldots B_{nn}b^1_{nn}),$$

or

$$\text{trace}\left[(\text{adj } B)\, B_\lambda\right] = \sum_{i,j=1}^{N} B_{ij}\frac{db_{ij}}{d\lambda}. \tag{19}$$

By comparing equation (17) and (19) we can readily see

$$(\det B)_\lambda = \text{trace}\left[(\text{adj } B)\, B_\lambda\right]. \tag{20}$$

By substituting equation (20) into the right hand side of (14) we have

$$(\text{adj } B)_\lambda = \frac{(\text{adj } B)\,\text{trace}\left[(\text{adj } B)\, B_\lambda\right] - (\text{adj } B)\, B_\lambda\, (\text{adj } B)}{\det B} \tag{21}$$

2. Summary or the Initial Value Problem.

The problem has been reduced to the solution of the system of ordinary differential equations for the adjoint of B and the determinant of B

$$(\det B)_\lambda = \text{trace}\left[(\text{adj } B)\, B_\lambda\right]. \tag{20}$$

$$(\text{adj } B)'_\lambda = \frac{(\text{adj } B)\,\text{trace}\left[(\text{adj } B)\, B_\lambda\right] - (\text{adj } B)\, B_\lambda\, (\text{adj } B)}{\det B} \tag{21}$$

The initial conditions for the system of nonlinear ordinary differential equations (20,21) are obtained by evaluating the determinant of B and the adjoint of B for $\lambda = 0$, or for any arbitrary initial value of $\lambda = \lambda_1$. Let us denote the initial conditions by

$$\text{adj } B(0) = B_0, \tag{22}$$

$$\det B(0) = b_0. \tag{23}$$

3. Numerical Integration.

The digital computer due to its iterative nature is the most effective means available for the solution of differential equations whose data are all specified at one point, a so called an initial value problem. The computer program is written in such a way that the Runge-Kutta method is used to determine the first few points and the rest of the points are evaluated by using the Adams-Moulton predictor-corrector formula, which cuts down considerably on the computer time.

Since the Runge-Kutta method of integration is written for the case when the left side of the differential equations is a column vector, and since the left side of our equation (21) is a matrix equation, we have to transform it into a

column vector form. This is accomplished by putting either all the columns or all the rows of the matrix in a consecutive column form (this means that if B is an n^2 matrix, then the column vector for (adj B) will have n^2 elements).

For example, if we have a square matrix

$$\text{adj } B = \begin{pmatrix} B_{11} & B_{21} & \cdots & B_{n1} \\ B_{12} & B_{22} & \cdots & B_{n2} \\ \vdots & & & \\ B_{1n} & B_{2n} & \cdots & B_{nn} \end{pmatrix} \tag{24}$$

its elements can be written in the following column form

$$B_{11} = Y_1 \ , \quad B_{21} = Y_{n+1} \ , \quad \ldots\ldots\ldots$$

$$B_{12} = Y_2 \ , \quad B_{22} = Y_{n+2} \ , \quad \ldots\ldots\ldots$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$B_{1n} = Y_n \ , \quad B_{2n} = Y_{2n} \ , \quad \ldots\ldots\ldots B_{nn} = Y_{(n^2)} \ .$$

The differential equation (20) is a scalar and equation (21) is a square matrix B of order $n \times n$. Therefore, equation (20) and the matrix differential equation (21) will yield us a system of $(1 + n^2)$ ordinary differential equations, which are readily solved with speed and accuracy on the computer.

4. Special Case of a Linear Eigenvalue Problem.

In many engineering problems one deals with the matrix given by

$$A - \lambda I, \tag{25}$$

where λ is the eigenvalue and I is a unit matrix. By introducing the notation

$$\mu = \frac{1}{\lambda} \tag{26}$$

in the above equation we deal with the following matrix

$$B(\mu) = I - \mu A \tag{27}$$

by differentiating (27) with respect to μ we obtain

$$B_{\mu}(\mu) = - A \tag{28}$$

The roots of the

$$\Delta = \det B(\mu) = 0 \tag{29}$$

or

$$\det (I-\mu A) = 0,$$

give the reciprocals of the eigenvalues λ .

Recalling equation (2), the system of the differential equations (20,21) become

$$\frac{d\Delta}{d\mu} = - \text{trace } (MA), \tag{30}$$

$$\frac{dM}{d\mu} = \frac{-M \text{ trace } (MA) + MAM}{\Delta}, \tag{31}$$

From equations (30,31) we see that it is convenient to introduce a new matrix C such that

$$MA = C \tag{32}$$

Then, equation (30) becomes

$$\frac{d\Delta}{d\mu} = - \text{ trace } C. \tag{33}$$

By post-multiplying both sides of equation (31) by A we have

$$\frac{d}{d\mu}\ (MA) = \frac{-MA \text{ trace } (MA) + MAMA}{\Delta} \tag{34}$$

using notation (32), equation (34) becomes

$$\frac{dC}{d\mu} = \frac{-C \text{ trace } C + C^2}{\Delta} \tag{35}$$

The initial conditions, from equation (29), for $\mu = 0$ are seen to be

$$\Delta(o) = \det (I) = 1. \tag{36}$$

From equation (27) we have

$$M(o) = \text{adj } I = I. \tag{37}$$

Then equation (32) gives

$$C(o) = A \cdot I = A \ , \quad C(o) = A. \tag{38}$$

In summary, the system of ordinary differential equations for determining μ are

$$\frac{d\Delta}{d\mu} = - \text{ trace } C, \tag{39}$$

$$\frac{dC}{d\mu} = \frac{- C \text{ trace } C + C^2}{\Delta} \tag{40}$$

with initial conditions

$$\Delta(o) = 1, \tag{41}$$

$$C(0) = A. \tag{42}$$

5. Methods for Locating the Eigenvalues.

We now will discuss three ways of employing the above relations.

If we know that the root is real (which is frequently the case) we may simply integrate along the real axis until the determinant becomes zero.

In the event that the roots are complex we may use some results from the theory of complex variables. The number N of zero of det.B(λ), assuming no poles, contained within a closed contour C is [5] ,

$$N = \frac{1}{2\pi i} \int_C \frac{\frac{d}{d\lambda}[\det B(\lambda)]}{\det B(\lambda)} \, d\lambda. \tag{43}$$

This may be evaluated numerically since we can integrate the differential equations (20, 21) around the contour C to produce values of the numerator and denominator of equation (43) and then use a numerical quadrature formula. In the event that there is only one root λ_1 , contained within the contour C, so that N=1, we may further use the formula

$$\lambda_1 = \frac{1}{2\pi i} \int_C \lambda \cdot \frac{\frac{d}{d\lambda}[\det B(\lambda)]}{\det B(\lambda)} \, d\lambda, \tag{44}$$

to precisely locate the root. The advantage of this is that we only require values on C and do not have to integrate

near a point for which the determinant Δ is near zero.

The formula (20) also suggests the use of Newton's method for finding the roots of equation

$$\Delta = \det B(\lambda) = 0.$$

Let λ_1, real or complex, be an approximate value of a root. Have the computer evaluate numerically the

$$\det B(\lambda_1), \quad \text{adj } B(\lambda_1) \quad \text{and} \quad B_\lambda(\lambda_1) .$$

Then, the next approximation λ_2 is, according to Newton's method,

$$\lambda_2 = \lambda_1 - \frac{\det B(\lambda_1)}{\frac{d}{d\lambda}[\det B(\lambda_1)]} \tag{45}$$

But, since from equation (20) we have

$$\frac{d}{d\lambda}[\det B(\lambda_1)] = \text{trace } \left\{M(\lambda_1) B_\lambda(\lambda_1)\right\} \tag{46}$$

then, using equation (46) in (45), we obtain

$$\lambda_2 = \lambda_1 - \frac{\det B(\lambda_1)}{\text{trace } \left\{M(\lambda_1) B_\lambda(\lambda_1)\right\}} \tag{47}$$

6. Numerical Example.

To illustrate this method on a nonlinear eigenvalue problem, consider the case of buckling of the frame shown in Fig. 1. By writing out the slope deflection equations for each member, taking into account the axial force in the columns and neglecting the axial force effect in the beams we obtain the following determinant for evaluation of the eigenvalues, where λ enters nonlinearly. The equations for the angles of rotation must satisfy the following, [1]

$$B(\lambda) = \begin{vmatrix} 4.5\ X(\lambda_2) + 1 & 1 & 0 \\ 1 & 2 & 0.5 \\ 0 & 0.5 & 4.5\ X(\lambda_1) + 1 \end{vmatrix}$$

where

$$\lambda_1 = k_1 \ell_1 = \ell_1 \sqrt{\frac{P}{EI}} = 0.5h \sqrt{\frac{P}{EI}} ,$$

$$\lambda_2 = k_2 \ell_2 = \ell_2 \sqrt{\frac{P}{2EI}} = \frac{h}{\sqrt{2}} \sqrt{\frac{P}{EI}} = \sqrt{2}\lambda_1$$

and

$$X(\lambda) = \frac{1}{9} \left(\frac{\lambda}{2}\right)^3 \frac{1}{\left[\tan \frac{\lambda}{2} - \frac{\lambda}{2}\right]}$$

The problem consists in determining the critical value of λ (i.e. P_{cr}), for which the buckling of the frame occurs first.

Solving the Cauchy system using the Runge-Kutta method of integration, the numerical result obtained by M. Scott by this method is

$$\lambda_2 = 4.11 .$$

This result is in perfect agreement with the solution obtained for this frame by the method of successive approximations as described in reference [2]. This example has been chosen in order to check the efficacy of the new method and to compare the obtained result against the existing approximate solution.

7. Conclusions.

An advantage of this method over other techniques is that it can readily be applied to solve nonlinear eigenvalue problems without the need of even expanding the determinant. The computer does all the matrix multiplications to evaluate the right hand side of equations (20,21) and then straightforward Runge-Kutta integration is done with known initial conditions. The values of λ for which the determinant becomes zero are the eigenvalues of the problem. We can determine the nonlinear eigenvalues for vibration problems as well as for problems of static and dynamic stability, which will be discussed in future papers.

REFERENCES

1. R. E. Kalaba, M. Scott, E. Zagustin, "A New Method for Nonlinear Matrix Eigenvalue Problems of Structural Mechanics". Submitted for publication and presentation at the Structural Dynamics meeting to be held in Las Vegas in April 1974.
2. A. S. Volmir "Stability of Elastic Systems" Fizmatgiz Moscow, 1967.
3. R. E. Kalaba, M. R. Scott, "An Initial-value Method for Integral Operators - IV. Complex-valued Kernels of Laser Theory". J. Quant. Spectrosc. Radiat. Transfer, Vol. 13, pp. 509-515, Nov. 1972.
4. R. Kalaba, M. Scott, " An Initial Value Method for Integral Operators II. Eigenfunctions", J. Optimization Theory and Applications, October, 1973.
5. R. Courant, Functionentheorie,Interscience, N.Y. 1945.

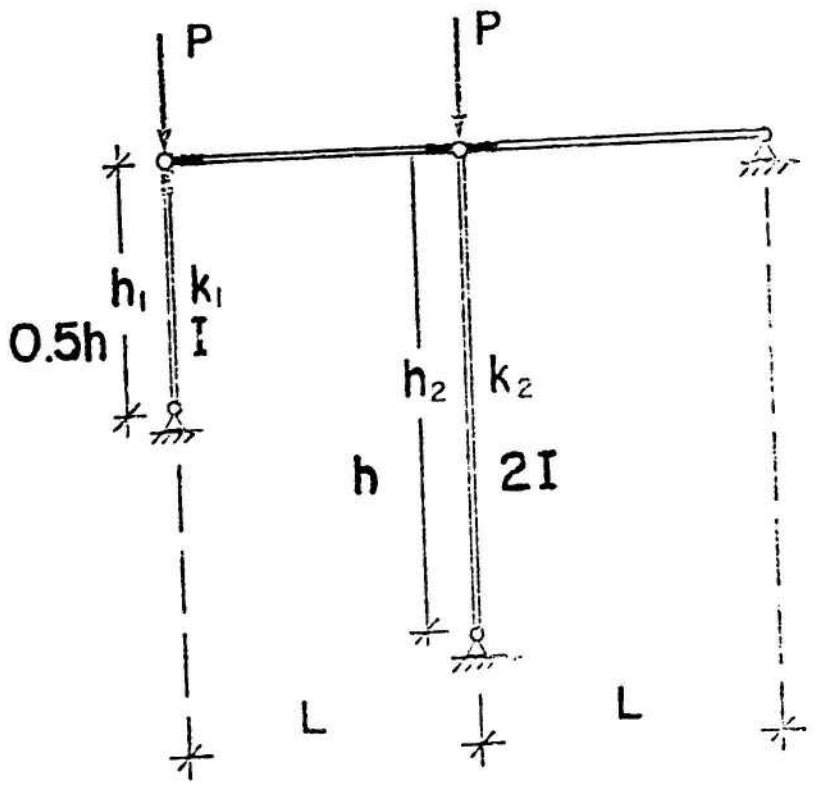


Figure 1

Nonlinear eigenvalue problem for buckling of a frame.

NUMERICAL SOLUTION SCHEMES FOR HIGHLY NONLINEAR STATIC STRUCTURAL RESPONSE

John F. McNamara
University of Illinois, Urbana,
and U. S. Army Construction Engineering Research Laboratory,
Champaign, Illinois

ABSTRACT. The feasibility and accuracy of a number of solution schemes for problems of static nonlinear structural response are studied. The structure is modeled by the finite element method, and the nonlinearities are mainly those of material behavior, but large displacement effects are also considered. It is shown that accepted solution approaches do not work as expected in the case of cyclic behavior of the structure around its limit load level. Results from two different finite element structural models are compared with those of an experiment for a simply supported beam undergoing load reversals at its midpoint. This simple example serves to illustrate the numerical problems encountered in analyzing engineering structures under severe lateral loads which initiate failure modes in the structure.

1. INTRODUCTION. The general problem under discussion is the analysis of framed structures under large reversals of applied loading. For the present the study is restricted to mild steel structures and to the effects of cyclic loading into the plastic range. The usual approach to this problem is called a second-order elastic-plastic analysis [1, 2, 3] where yielding occurs as a plastic hinge of zero length at points of maximum moment such as joints between beams and columns and concentrated load points. A bilinear or multilinear elastic-plastic moment-rotation relation is assumed for these points, and one can proceed within the usual assumption of the stiffness approach where the formation of a plastic hinge means a new stiffness matrix is assembled and the results form a series of multilinear responses. More refined results may be obtained by continuously updating the stiffness matrix. The complexity of the solution depends upon the material behavior model, and numerical solutions for load reversals appear feasible. Experimental results given in [3] for a simple frame structure under cyclic loading show a 40 percent increase in maximum lateral load capacity over the load predicted by second-order elastic-plastic analysis. This difference indicates the necessity of performing a more complete cyclic analysis in order to determine the energy absorption capacity of the frame more accurately. An overview of the variety of load-displacement hysteresis loops for frame type structures, where buckling, plastic straining, large displacements and joint slip are active is given in [4].

The drawback with an analysis based on the plastic hinge concept is that the moment-rotation curve must be known for an indefinite number of cycles and plastic strain amplitudes. In all cases known to the writer, cyclic material behavior has been deduced from monotonic test curves and is, therefore, skew-symmetric about the origin of the moment and rotation axes [1]. On the other hand, refined material models, expressed as stress-strain relations and incorporating cyclic effects in a highly accurate manner, are available [5, 6].

Since these relations depend on stresses and strains at a point rather than generalized variables, a finite element rather than a stiffness approach is used to discretize the structure. A start in this direction has already been accomplished in [7] where a plane stress finite element beam model is used to obtain the cyclic behavior of a simply supported mild steel beam loaded at its center. This work and experimental values for the above problem will be discussed later in this report.

A surprisingly minute amount of finite element literature, germane to the problem at hand, can be found by the writer. A vast body of work, relating to elastic-plastic solutions of one and two dimensional stress systems, exists, but only for the case of monotonic loading up to the limit or collapse load. An excellent survey of this research area is given in [8] where all the standard solution procedures are developed and commented on. One of the preferred schemes is defined as a first-order self-correcting method. In this case, linear incremental forms of the governing set of nonlinear equations are used with the unbalance in equilibrium forces, computed at the end of every increment, being applied as a corrective load on the following increment. Convergence with solution schemes such as the above is usually obtained by relative comparisons of responses with diminishing sizes of load increments. Such methods appear to be highly accurate for establishing monotonic load-displacement relationships, but it can be stated that the solution error builds rapidly as the limit load of the structure is approached. For cyclic loading into and out of the limit load range, the error control has to be more rigorous, and equilibrium must be satisfied at all steps if possible.

With the exception of [7] only one other group has included cyclic loading conditions in solving general nonlinear problems using the finite element method. This work has been carried on over a number of years and is summarized in [9]. The load reversals considered are at levels below the limit load for the example problems, and no numerical difficulties were encountered except in one case with the largest absolute value of load. No computation is made on the equilibrium of forces at any stage of the solution with reversals, but the results look very reasonable for the problems attempted. All solutions were load controlled as opposed to specifying displacement increments.

In summary, no real test has yet been carried out in order to extend many accepted solution procedures for nonlinear structural equations to the more general problems of reversed loading over an indefinite number cycles. Preliminary results of the writer's work in this regard are presented in the following, and comparisons are made with similar work given in [7].

2. FUNDAMENTAL EQUATIONS. The finite element formulation of equilibrium equations for problems of nonlinear structural mechanics has been detailed at length in many publications of which [8, 10, 11] are pertinent to the present study. For the purposes of identifying the basis of a particular solution scheme, the equations will be reviewed briefly here. The presentation is also simplified by assuming that strains remain small and that only moderately large rotations of members occur.

The general equation of equilibrium in a Lagrangian reference frame is obtained as

$$\int_V [B]^T \{\sigma\} dV = \{P\} \tag{1}$$

where $\{\sigma\}$ is a vector of generalized Kirchoff stresses, $[B]$ transforms generalized displacement increments at the nodes to generalized strain increments in the body, and $\{P\}$ includes load contributions from nodal loads and distributed pressures on an element.

The matrix $[B]$ contains nonlinear terms of quadratic order in the displacement increments. For the one dimensional beam problems to be considered here, it is based on the expression

$$\Delta E_x = \frac{d(\Delta u)}{dx} + \frac{du}{dx}\frac{d(\Delta u)}{dx} + \frac{1}{2}\left(\frac{d(\Delta u)}{dx}\right)^2 + \frac{dv}{dx}\frac{d(\Delta v)}{dx} + \frac{1}{2}\left(\frac{d(\Delta v)}{dx}\right)^2 - z\frac{d^2 v}{dx^2} \tag{2}$$

where u is along, and v perpendicular to, the beam axis x, and z is the distance of the beam fiber from the neutral axis. In the above ΔE_x is a Lagrangian strain measure, and nonlinear terms for both u and v deformations have been included although the former could be discarded since their influence is minimal. Nonlinear terms are not included for the curvature strain. The quadratic terms in (2) insure, for a nonlinear elastic system at least, that stresses calculated as the sum of successive incremental values during the solution process will equal the stress calculated from the total displacement at any state of the solution. This refinement is important when it becomes necessary to make equilibrium checks on the solution [12].

The elastic-plastic constitutive equation is introduced through the linear incremental relation

$$\{\Delta\sigma\} = [D]\{\Delta\varepsilon\} \tag{3}$$

where $\{\varepsilon\}$ is a vector of generalized strains and $[D]$ is constructed in the manner outlined in [13].

In order to set up a basis for the commonly used solution schemes, a first-order expansion of (1) is made about some known equilibrium state giving

$$\int_V [\Delta B]^T\{\sigma\}\, dV + \int_V [B]^T\{\Delta\sigma\}dV = \{\Delta P\} + (\{P\} - \int_V [B]^T\{\sigma\}\, dV) \quad (4)$$

Following techniques outlined in [14], one can form a stiffness matrix with the terms on the left hand side of (4) and rewrite it as

$$[K]_T\ \{\Delta q\} = \{\Delta P\} + \{I\} \quad (5)$$

where $\{\Delta q\}$ is the vector of generalized or nodal displacements. $\{I\}$ is the vector of forces found from the terms in parentheses in (4) and is zero if equilibrium is exactly satisfied at the beginning of the current increment. The matrix $[K]_T$ is a nonlinear tangential stiffness matrix

defined for the elastic-plastic case as

$$[K]_T = \int_V [B]^T[D][B]dV \quad (6)$$

3. SOLUTION PROCEDURES. Historically, the basis for solving nonlinear structural problems was equation (5) without the vector $\{I\}$ of unbalanced forces. This is a simple linear incremental, or marching, process and while it is the most economical of all methods, it is also the least accurate. When (5) is used as presented, the process is named "load-correction," or "first-order self-correction." It is unlikely that either of these methods will be suitable for use where load cycling occurs since there is no error control. Monotonic loading is well represented by (5) although very small increments are required for any but the simplest problems [8]. Both methods may give reasonably approximate values for limit or collapse loads within a relatively small number of increments.

It appears necessary to use some iterative process where the order of the error can be specified to any desired degree. The most accurate approach for any general case is the Newton-Raphson process, and we again use (5) as a starting point. For the first iteration one solves (5) which is now rewritten as

$$[K]_T\{\Delta q\}_1 = \{\Delta P\} + \{I\}_o \quad (7)$$

where $\{I\}_o$ is calculated at the beginning of the increment and is usually

zero, or very close to zero. Using $\{\Delta q\}_1$, one can now calculate $\{I\}_1$, or

the unbalanced or "residual" force, at the end of the increment. Then, changes in the displacement increment can be calculated as follows:

$$[K]_T\{\Delta(\Delta q)\}_i = \{I\}_{i-1} \qquad i = 2, 3, \ , n \quad (8)$$

where i indicates the number of iterations. The displacement over the increment at any stage is found from the relation

$$\{\Delta q\} = \{\Delta q\}_1 + \sum_{i=2}^{n}\{\Delta(\Delta q)\}_i \quad (9)$$

and can be used to find a current value for $\{I\}_{i-1}$. If the nonlinear stiffness matrix $[K]_I$ is updated for every solution of (8), the process becomes an iterative incremental Newton-Raphson solution of the equation of equilibrium (1), i. e.,

$$\{I\}_{i-1} = 0. \tag{10}$$

The iterations are continued until $\{I\}_{i-1}$ is arbitrarily close to zero, or insignificant changes in displacement increments are found.

The continual updating of the tangent stiffness can become very expensive in computer time, and many modified forms of the full Newton-Raphson process have been proposed [10, 11]. The simplest approach is to use the linear elastic tangent stiffness matrix $[K]_E$ in (7) and (8) throughout the solution. This method is called the "constant stiffness," or "initial stress," method and examples of its use are given in [11, 15]. It is shown in [16] that the process is convergent except in some large displacement problems and near failure, or limit loads, in plasticity problems. In cases involving a high degree of nonlinearity, the convergence is very slow and time consuming. An advantage of the "constant stiffness" method is its ability to follow a softening load displacement response. This is due to the positive definite character of the linear elastic coefficient matrix used throughout the solution.

An alternative, but very similar approach, to the "initial stress" method is the method of "initial strain," or "thermal strain." This approach preceeded the "initial stress" method and has been applied to many engineering problems [8, 18], but is not considered in the present study.

Variations on the complete Newton-Raphson approach are also obtained by selective updating of the stiffness matrix. Procedures in common use are to update in each increment for the first iteration only or after two or three iteration cycles [10, 11]. In all cases the stiffness is reassembled once per increment only, and the purpose is to obtain a reasonably accurate and economical solution. This approach is very suitable for all but the highly nonlinear problems.

As a postscript to this discussion of solution schemes, it is pointed out that, in the general case, there is no guarantee of convergence or uniqueness with any of the methods.

4. APPLICATION TO CYCLIC LOADING. It is reasonable to state that no one solution scheme for general nonlinear structural problems will be completely satisfactory. A balance must always be maintained between the competing constraints of cost and accuracy since they are usually directly related to each other. The order of the solution scheme required is a function of the degree of nonlinearity in the response, and numerical experiments must be made in order to select a suitable scheme for the class of problem under consideration. The elements of the current problem are shown in Fig. 1 where a simply supported beam of A36 steel is

loaded through one complete cycle with displacements large enough to cause substantial alternating plastic strains in the beam fibers. The experimental load-displacement curve is characteristic of the hysteresis loops obtained by testing framed structures under lateral loads applied at the floor levels. There is a steep elastic first stage followed by a very flat portion once a mechanism forms. In this particular case, since the beam is statically determinate, the mechanism stage begins as soon as plastic yielding extends through the midspan section and forms a plastic hinge. In more highly statically indeterminate framed structures, the change from elastic to mechanism behavior is less abrupt due to the successive development of plastic hinges throughout the structure. In any case, the initial response can be divided into an elastic section, a transition section, and a flat mechanism section. The latter two sections create numerical difficulties and demand a refined, rather than an approximate, solution method.

The unloading from the first mechanism stage and subsequent cycles appear smoother than the initial phase, and this is due to the work-hardening of A36 structural steel. For idealized material behavior the reloading curves will have the same appearance as the initial branch, and in fact could be constructed from knowledge of the initial branch alone [9].

The finite element models which are used in this study are illustrated in Fig 2. The plane stress element formed the basis of the structural model in [7]. The element is parabolic in order of displacements and belongs to the "Serendipity" element family [19]. The stiffness was assembled by numerical integration based on a 3 x 3 Gaussian quadrature rule. In the interest of converting from a continuum model to a structural model of the beam, the simple beam bending element was incorporated into the study. There are three degrees of freedom per nodal point, two translations, and one rotation. The displacement approximation is cubic in the transverse direction and linear in the axial direction. Numerical integration along the length of an element is carried out at three Gauss points, and the integration through the depth is obtained by Simpson's Rule using 6 intervals or layers.

5. CHARACTERISTICS OF SOLUTION SCHEMES. The performance of the various solution schemes were investigated by attempting to simulate numerically the initial branch of the load-displacement curve shown in Fig. 1. It was decided to carry the solution just beyond the first elastic unloading with the assumption that numerical problems typical of a complete cyclic analysis would be encountered therein.

The beam was descritized by using 10 beam elements for the half-span. For this test problem the material model is that shown in Fig. 1 with the further assumptions of isotropic hardening and von Mises' yield criterion.

Numerical results for the solution schemes under consideration are given in Fig. 3. The loading program for solutions A, B, and C was the same and required a certain amount of trial and error before stable results were obtained. Twenty load increments of magnitudes varying from 200.0 (elastic) to 5.0 lbs. were used to reach the 570 lb. level, and 2.5 lb. increments were used for the remainder of the response. For the iterative solutions, convergence was assumed to have occurred when the following

inequality was satisfied:

$$\frac{\|\{\Delta(\Delta q)\}_i\|}{\|\{\Delta q\}\|} \leq 0.001 \qquad (10)$$

The above tolerance was deduced from observing typical values associated with essentially zero residual, or unbalanced, equilibrium forces in the beam.

The Newton-Raphson process was by far the most efficient of the three load controlled schemes. The "initial stress" approach followed the Newton-Raphson curve until it had to be terminated due to excessive cost. As the plastic collapse load was approached, the number of iterations in an increment began to increase rapidly as shown in Fig. 3. The direct self-correcting method is the least costly, but the internal reaction at midspan was computed as -241.42 lbs., which indicates an error of 66.5 percent with respect to the applied half-span load of 720.0 lbs. The midspan displacement value is also noted to have substantial errors at the time of unloading. A buildup of errors of this order in successive cycles would render meaningless results. The initial elastic unloading is also not correct due to the large residual error in forces. When computer running times are normalized with respect to the Newton-Raphson solution, the cost for solution A was 0.475, and solution C had already cost 1.1 units at termination.

For the loading program selected, the Newton-Raphson solution satisfied equilibrium exactly at all steps, and the average number of iterations per increment was four. A problem noted with this method is its inability to remain stable for perfectly-plastic material behavior or small values of the hardening modulus. As an example, the current beam problem was attempted with a hardening modulus of 100,000 p.s.i., and a stable solution beyond the collapse load was not obtainable. Convergence failures of this type were also noted in [8, 9, 20]. For this reason, and in anticipation of the fact that load control fails for load displacement curves with negative slopes, it was decided to attempt a displacement controlled Newton-Raphson solution.

The convergence of this latter solution is shown for half of one complete cycle in Fig. 4. The sensitivity of the solution to displacement increment size is indicated by the uneven, and sometimes unstable, responses obtained with the trial values. The subscripts 1, 2, and 3 on the displacement increment values distinguish the successive attempts made to compute a particular section of the response curve with diminishing increment sizes. The part of the response up to the first elastic unloading in Fig. 3 is superimposed on Fig. 2 as plot D. In actual fact, D was obtained initially and indicated a suitable load program for the other curves. Although equilibrium was satisfied to the same tolerance with solutions D and B, a difference in the response occurs at the knee due to the larger displacement increments used. This difference is carried forward after the displacement controlled solution stabilizes.

However, a somewhat more important feature of these solutions is their rate of convergence characteristics as given by the values listed in Table 1. The results shown apply to the final points on the curves before elastic unloading. The comparison is also fair to both solutions in that increment sizes were equivalent in load and displacement over the response after the knee section. It is apparent that the load controlled solution converges more than twice as fast as the displacement controlled version, and this was found to be consistently true over the complete response. With load control the value of the applied load is fixed, and this fixes the reaction at the support point so that only the internal force at the midspan must be equalized with the externally applied load. When a displacement increment is specified, both the support reaction and midspan force vary with each iteration, and this possibly creates a more difficult situation for obtaining an equilibrium set of forces. However, this convergence phenomenum is diametrically opposite to the behavior noted in [10] for nonlinear elastic systems.

An odd feature with solution D is the recurring equilibrium of the boundary forces on the beam as shown by the underlined values in Table 1. Internal forces are not in equilibrium, however, so the iteration continues. This behavior indicates the necessity of checking global equilibrium of forces or using a displacement convergence criterion instead. The cost factor for solution D was 1.3, and a total of 34 increments were used as opposed to 46 with solution B.

The nonlinear geometric stiffness contribution from the higher order terms in equation (2) were included in the analysis using the approach already described for curve A of Fig. 3. As expected, the maximum beam displacement of 0.025 inches is too small for any large displacement effects to be evident. In fact, the results are, practically speaking, exactly equal to those of curve A. The rate of convergence changes, however, and many redundant iterations were performed since equilibrium was satisfied at tolerances ranging between values of 0.02 to 0.005 for inequality (10).

6. COMPARISONS WITH PREVIOUS RESULTS. The numerical results, derived in [7] for the same problem using the plane stress element of Fig. 2, and experimental results [7] are compared with those of the writer from Fig. 4. It should be pointed out that the beam dimensions made it more suitable for a plane stress rather than a bending analysis. The dimensions were dictated by the fact that large plastic strains were required without large displacements since the influence of change in geometry was not included in [7]. An interesting feature of the plane stress analysis is that only six load increments were used to reach the first unloading stage. The method was a modified Newton-Raphson process where the stiffness is updated at every third iteration in any one increment. The average number of iterations was 6, but the final point before the first unloading required 15 iterations. The tolerance specified that the ratio of the norm of the residual force vector to that of the applied load vector be less than 0.01.

Also, the cost factor with respect to solution A of Fig. 3 is 0.25 approximately. This is explained by the fact that the simpler beam model has 210 (7x3x10) stress points as compared with only 135 (3x3x15) for the plane stress element. The relative ease with which this solution was obtained is explained by the fact that a continual redistribution of the two-dimensional stress system is taking place resulting in an increasing load capacity, and a limit load, as with axial stresses only, is not reached. However, bending alone is more typical of structural components such as I-beams.

Since the problem under study is a thick beam under a point load, one can expect the lower load levels as shown for the bending analysis in Fig. 5. The writer's results were not continued beyond those shown since they are based on isotropic, rather than kinematic, hardening. This leads to the greater elastic unloading range over the plane stress analysis and also, of course, to lower load values on the reyielded portion of the response. It is apparent that agreement is such that one can have confidence in the solution schemes which is the primary goal of the current study.

7. CONCLUSIONS. A contribution of this study is that it effectively eliminates the "initial stress" and the "first-order self-correcting schemes for future consideration in problems of cyclic loading. Cost and accuracy are the deciding factors, respectively, with these two methods.

The Newton-Raphson process with load control appears best, but its limited applicability must be considered a serious drawback. Subsequent to the calculations shown in the figures and in Table 1, it was noted that the tangent stiffness was only reassembled in an increment after the first two steps. This explains the jump in error measure between the first and second iterations in Table 1 (from 0.327 to 0.942) and the equilibrium of forces in iteration 2. In other words, there is a superfluous iteration in the load control results whose removal will even further improve the efficiency of this approach. The same will not hold true for the displacement controlled Newton-Raphson since its convergence rate appears somewhat arbitrary.

In investigating the effect of geometry changes on the response of solution A, it was noted that their inclusion changed the convergence rate in a favorable manner by allowing an increase in the error tolerance from 0.001 to 0.005 approximately. The same has not, as yet, been attempted with the displacement controlled solution, but a beneficial effect on solution convergence can be expected.

While the numerical techniques tested in this study indicate that convergent solutions can be obtained, in the long run, no one of the methods is fully satisfactory. Newton-Raphson with load control will fail for structures with a flat or softening load-displacement response, and displacement control is only suitable for fundamental investigations of experimental structures since, in reality, a structure is specified as acting under a set of loads. The basis for an alternative approach,

which is general in all respects, is given in [21]. The features of the method are that the constant elastic stiffness is used in solving the equations and the correction for nonlinear effects is made to the displacement increments in an external fashion. The method could be classified as the "initial stress" method with a higher order correction. A further improvement in convergence rate may be possible by adopting a new formulation for the elastic-plastic constitutive relations which is given in [22]. Thermal strain cycling of pressure vessel components was treated successfully in [22] using a finite difference structural model. These latter refinements will be applied to cyclic loading of frame structures in the near future.

ACKNOWLEDGEMENT. The research reported herein was carried out at the Materials Systems and Science Division, U. S. Army Construction Engineering Research Laboratory, under the supervision of Dr. Walter E. Fisher, Chief, Structural Mechanics Branch. The writer is grateful for the research support provided by the above institution.

REFERENCES

1. Popov, E. P., and Pinkney, R. B., "Reliability of Steel Beam-to-Column Connections under Cyclic Loading", Proceedings of the Fourth World Conference on Earthquake Engineering, Santiago, Chile, January 13 - 18, 1969.

2. English, G. W., and Adams, P. F., "Experiments on Laterally Loaded Steel Beam-Columns", Journal of the Structural Division, ASCE, Vol. 99, No. ST7, July, 1973, pp. 1457 - 1470.

3. Carpenter, L. D., and Lu, Le-Wu, "Repeated and Reversed Load Tests on Full-Scale Steel Frames", Proceedings of the Fourth World Conference on Earthquake Engineering, Santiago, Chile, January 13 - 18, 1969.

4. Wakabayashi, M., "Frames Under Strong Impulsive, Wind or Seismic Loading", Proceedings of the International Conference on Planning and Design of Tall Buildings, Lehigh University, Bethlehem, Pennsylvania, August 21 - 26, 1972.

5. Morrow, Jo Dean, " Cyclic Plastic Strain Energy and Fatigue of Metals", Internal Friction, Damping and Cyclic Plasticity, ASTM, STP-378, 1965, P. 77.

6. Martin, John F., "Cyclic Mechanical Tests and an Appropriate Analytical Stress-Strain Model for A36 Steel", Report of the Construction Engineering Research Laboratory, Army Corps of Engineers, Champaign, Illinois, 61820 (In draft).

7. Plummer, F. B., "A New Look at Structural Energy Dissipation", Report of the Construction Engineering Research Laboratory, Army Corps of Engineers, Champaign, Illinois, 61820 (In draft).

8. Stricklin, J. A., Haisler, W. E., Von Riesemann, W. A., "Formulation, Computation, and Solution Procedures for Material and/or Geometric Nonlinear Structural Analysis by the Finite Element Method", Report SC-CR-72 3102, Sandia Laboratories, Albuquerque, New Mexico, July, 1972.

9. Armen, H., Levine, H. S., Pifko, A. B., "Plasticity-Theory and Finite Elemer Applications", Proceedings of 2nd U. S. - Japan Seminar on Matrix Methods of Structural Analysis, University of Alabama Press, Huntsville, Alabama, August, 1972. pp. 393 - 437.

10. Zienkiewicz, O. C., and Nayak, G. C., "A General Approach to Problems of Large Deformation, and Plasticity Using Iso-parametric Elements", Proceedings of the 3rd Conference on Matrix Methods in Structural Mechanics, Wright-Patterson AFB, October 19 - 21, 1971.

11. Nayak, G. C., and Zienkiewicz, "Elasto-Plastic Stress Analysis. A Generalization for Various Constitutive Relations Including Strain Softening", International Journal for Numerical Methods in Engineering, Vol. 5, 1972, pp. 113 - 135.

12. Hibbitt, H. D., and Marcal, P. V., "A Numerical Thermo-Mechanical Model for the Welding and Subsequent Loading of a Fabricated Structure" Brown University Technical Report, N00014-006/2, March 1972

13. Marcal, P. V., "Finite Element Analysis with Material Nonlinearities--- Theory and Practice", Proceedings of 1st Japan - U. S. Seminar on Matrix Methods of Structural Analysis and Design, August 25 - 30, 1969, Tokyo, Japan.

14. Marcal, P. V., "Finite Element Analysis of Combined Problems of Nonlinear Material and Geometric Behavior", Division of Engineering, Brown University, Report N00014-007/2, June, 1969.

15. Zienkiewicz, O. C., Valliappan S., and King, I. P., "Elasto-Plastic Solutions of Engineering Problems; Initial Stress Finite Element Approach", International Journal of Numerical Methods in Engineering, Vol. 1, 1969, pp. 75 - 100.

16. Nayak, G. C., "Plasticity and Large Deformation Problems by the Finite Element Method", Ph. D. Thesis, University of Wales, Swansea, 1972.

17. Gallagher, R. H., Padlog, J., and Bijlaard, P. P., "Stress Analysis of Heated Complex Shapes", ARS Journal, May, 1962, pp. 700 - 707.

18. Witmer, E. A., and Kotanchik, J. J., "Progress Report on Discrete-Element Elastic and Elastic-Plastic Analysis of Shells of Revolution Subjected to Axisymmetric and Asymmetric Loading", Proceedings 2nd Conference on Matrix Methods in Structural Mechanics, Wright-Patterson AFB, Dayton, Ohio, October, 1968.

19. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill, London, 1971.

20. Isakson, G., Armen, H., Jr., and Pifko, A., "Discrete-Element Methods for the Plastic Analysis of Structures", NASA Contractor Report, NASA CR-803, October, 1967.

21. Nayak, G. C., and Zienkiewicz, O. C., "Note on the "Alpha'-Constant Stiffness Method for the Analysis of Non-Linear Problems", International Journal for Numerical Methods in Engineering, Vol. 4, 1972, pp. 579 - 582.

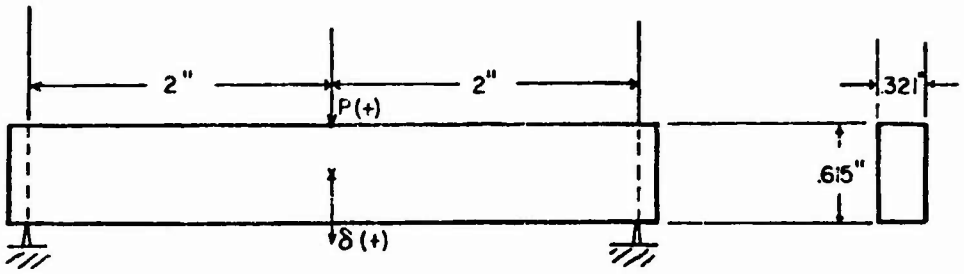
22. Barsoum, R. S., "A Convergent Method for Cyclic Plasticity Analysis with application to Nuclear Components", International Journal for Numerical Methods in Engineering, Vol. 6, 1973, pp. 227 - 236.

TABLE I

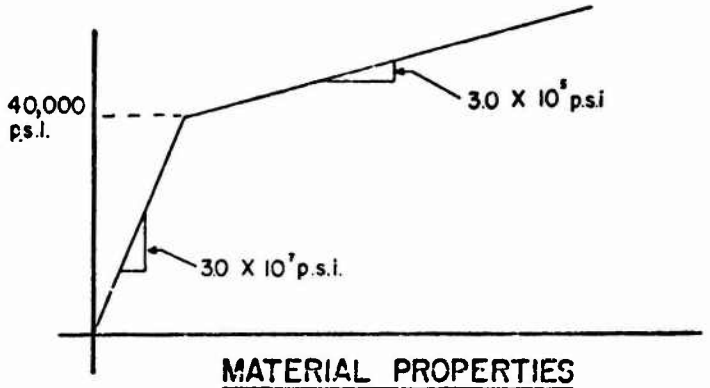
CONVERGENCE CHARACTERISTICS OF LOAD AND

DISPLACEMENT CONTROLLED NEWTON-RAPHSON METHODS

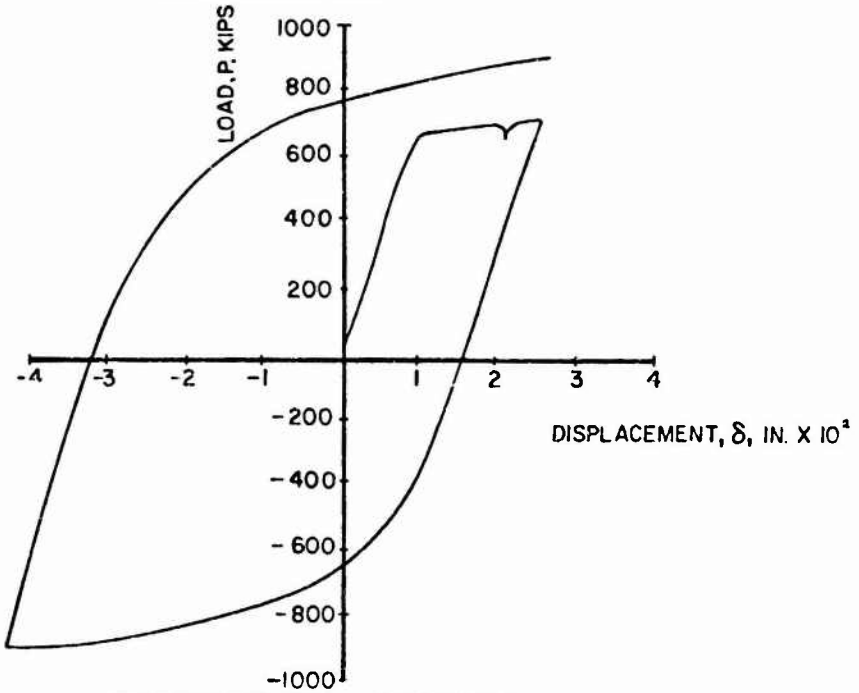
	LOAD CONTROL			DISPLACEMENT CONTROL		
ITERATION	INTERNAL FORCE		ERROR	INTERNAL FORCE		ERROR
NUMBER	SUPPORT	MIDSPAN	MEASURE	SUPPORT	MIDSPAN	MEASURE
0	647.5	-645.03	-	673.3404	-643.7756	-
1	647.5	-645.05	0.327	661.6573	-643.9652	0.05360
2	647.5	-647.5	0.942	645.5255	-645.5255	0.21220
3	647.5	-647.5	0.27×10^{-7}	644.4235	-541.5684	0.13035
4				631.3242	-631.3242	0.13593
5				625.0382	-593.7156	0.05251
6				617.7793	-617.7993	0.04786
7				616.5497	-616.5497	0.00568
8				616.5497	-616.5497	0.88×10^{-13}



BEAM EXAMPLE

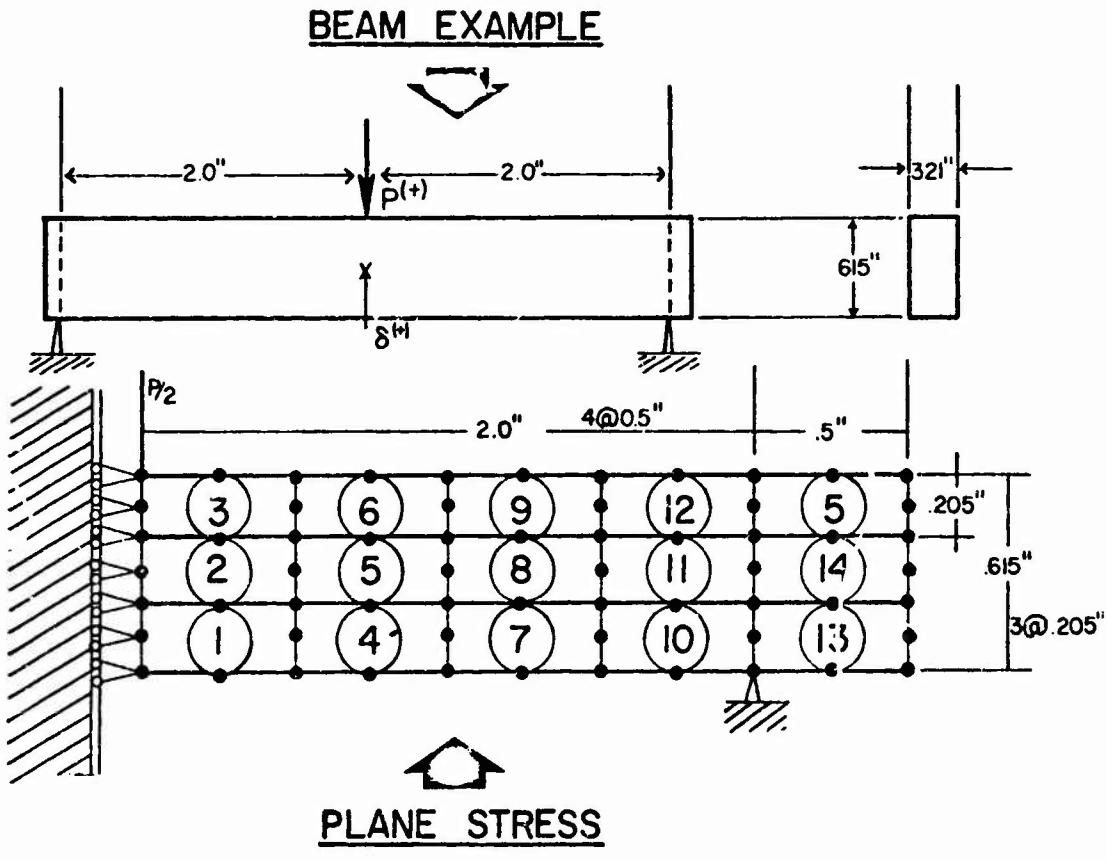


MATERIAL PROPERTIES



EXPERIMENTAL PROGRAM

FIG. 1



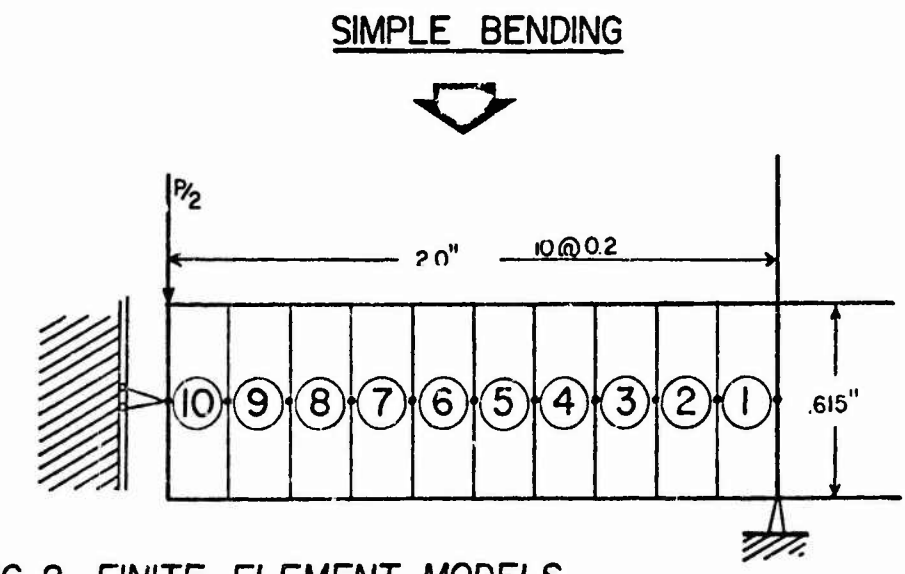


FIG 2 FINITE ELEMENT MODELS

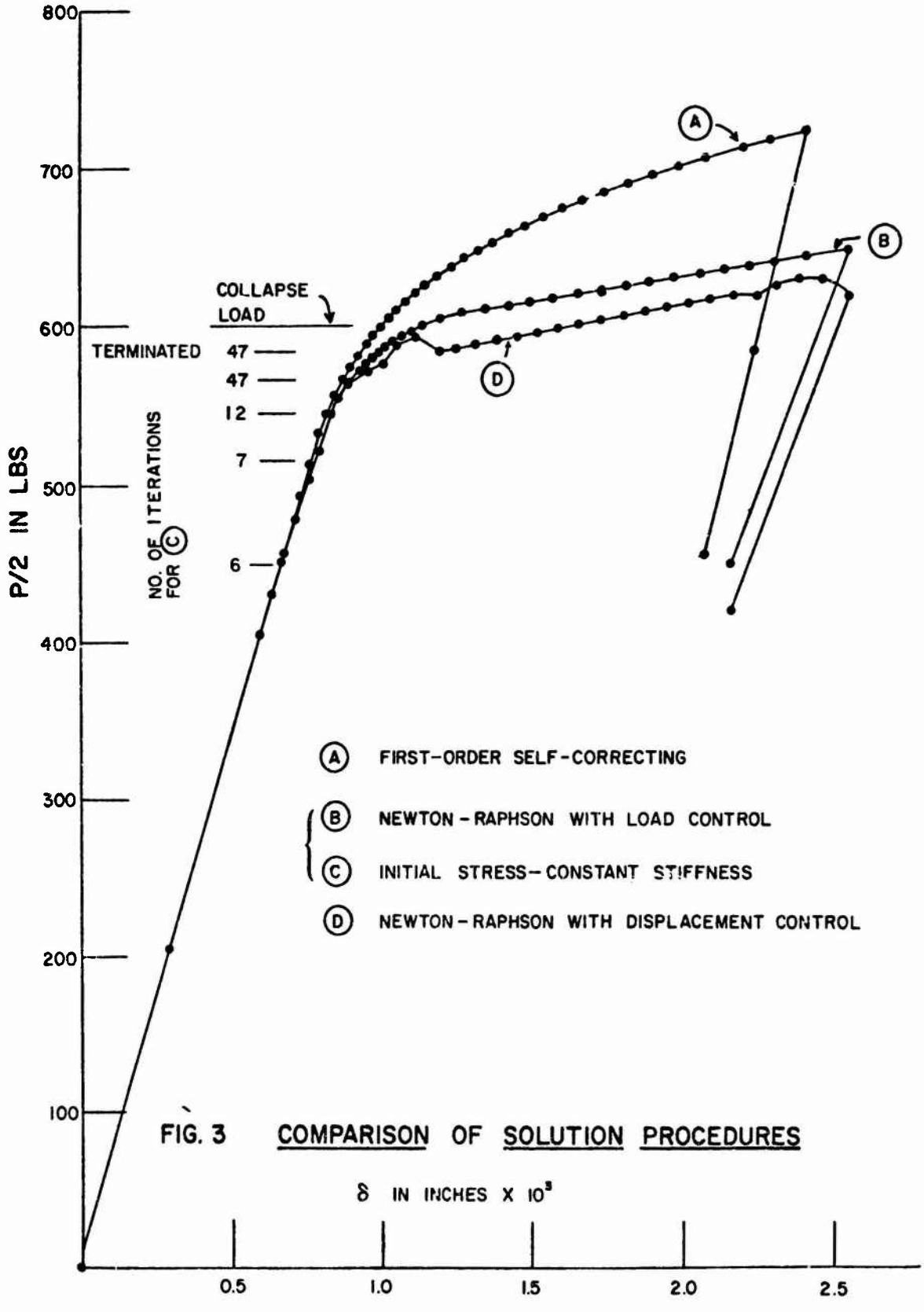


FIG. 3 COMPARISON OF SOLUTION PROCEDURES

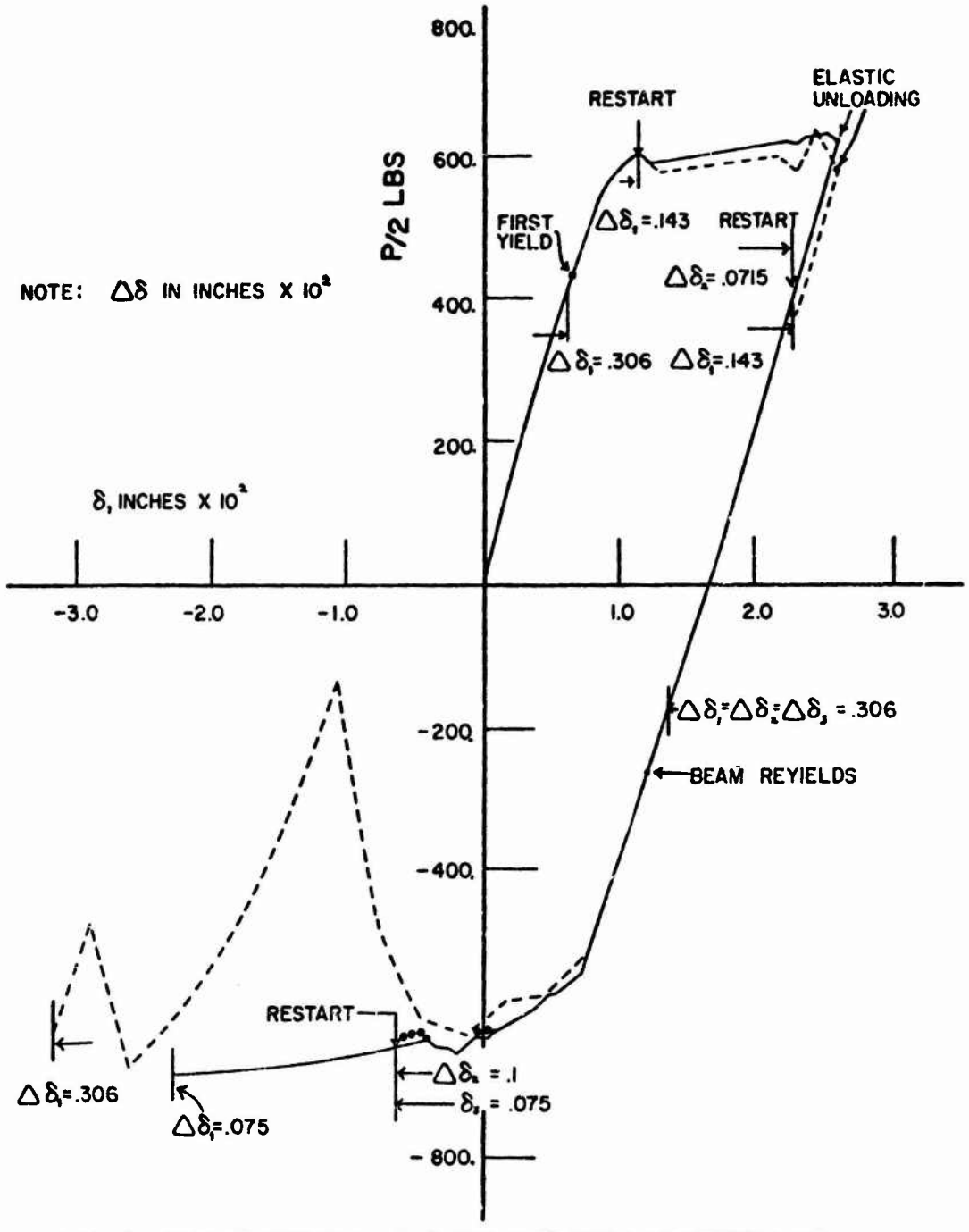


FIG. 4 CONVERGENCE OF DISPLACEMENT CONTROLLED NEWTON = RAPHSON SOLUTION

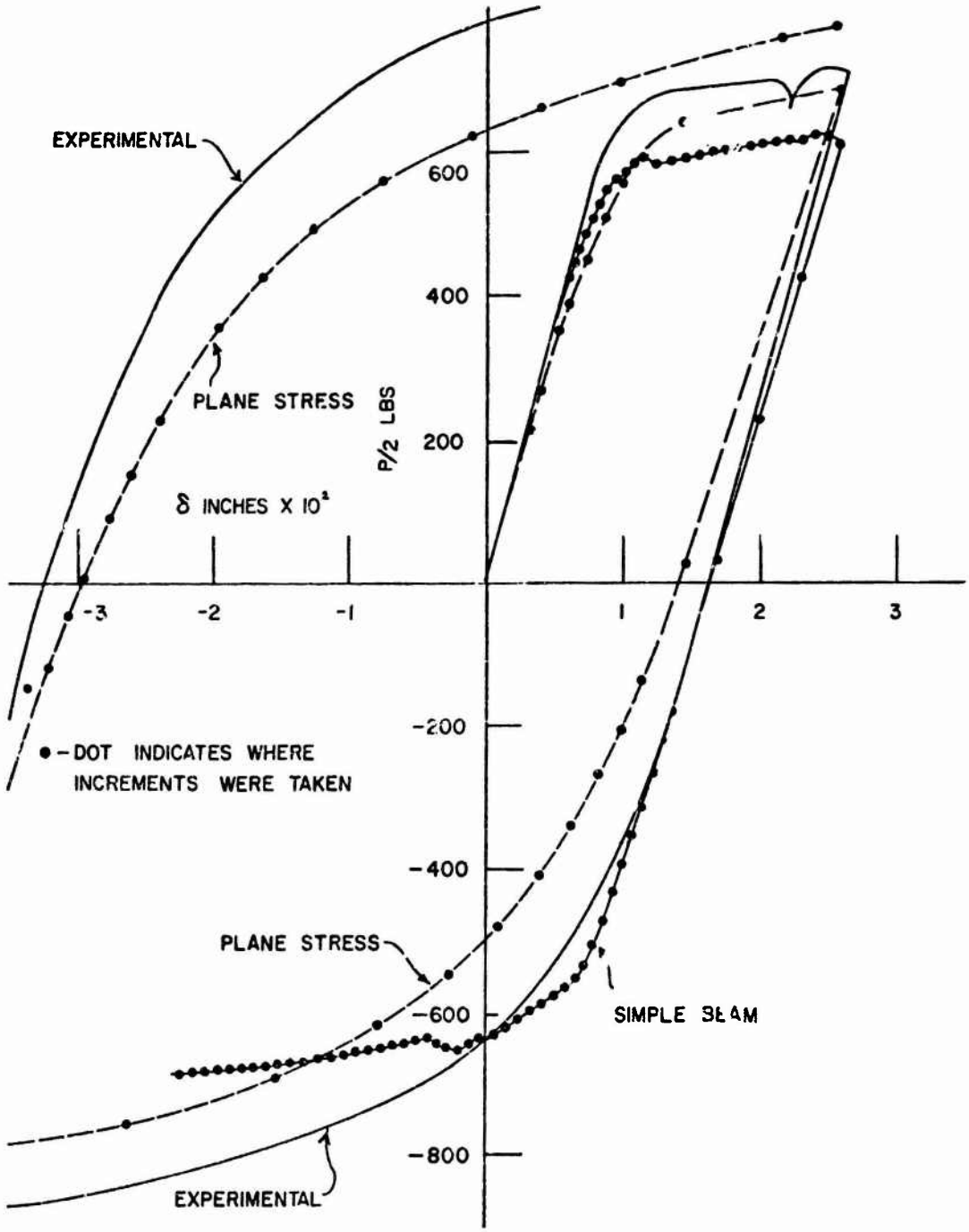


FIG. 5 COMPARISON OF PLANE STRESS AND SIMPLE BEAM FINITE ELEMENT RESULTS WITH EXPERIMENTAL VALUES.

DEVELOPMENT OF NUMERICAL METHODS FOR THE VELOCITY AND TEMPERATURE DISTRIBUTION IN AXISYMMETRIC SOLIDS UNDERGOING LARGE PLASTIC DEFORMATION

Taylan Altan
Battelle - Columbus Laboratories
Columbus, Ohio

and

Paul Gordon
Materials Engineering Division
Pitman-Dunn Laboratory
Frankford Arsenal
Philadelphia, Pennsylvania

ABSTRACT. Metal forming technology plays an important role in the manufacture of much of the Army's materiel. At Frankford Arsenal, for example, a major emphasis in current development programs for improved material artillery shell and cartridge cases is in developing new or improved forming processes. It is the purpose of this paper to present some recent advances in the development of computerized methods which provide consistent, approximate solutions to the field equations governing metal forming processes. Applications to extrusion, drawing and compression are given.

Part of this work was performed at Frankford Arsenal under the task for Mathematical Modeling of Forming Processes and part was performed at Battelle Columbus Laboratories under internal research funding. The authors have collaborated on the preparation of this presentation.

The plastic deformation of metals involves large irrecoverable strains. The work of deformation appears essentially in the form of heat energy. The determination of temperatures in a plastically deforming metal is basically a problem of time dependent heat flow in an incompressible moving medium with heat generation in the medium. In axially symmetric flow the equations governing the velocity and temperature distributions are found to be a set of coupled, severely nonlinear partial differential equations. In this study a combination of a numerical extremum method and a finite difference procedure were simultaneously employed to solve these equations. A velocity field, with undetermined coefficients, was constructed to identically satisfy certain of the field equations (incompressibility and continuity) and auxiliary conditions. Based on this field, the deformation energy functional was calculated. The undetermined velocity coefficients were determined by requiring the energy functional to be a minimum. Minimization was performed numerically. A finite difference approximation was applied iteratively to calculate the resultant temperature distribution.

This overall numerical algorithm was found to be relatively simple and efficient. These numerical techniques were applied to predict temperature distributions in axisymmetric deformation problems such as extrusion, wire drawing, and compression.

1. INTRODUCTION. The purpose of this paper is to describe the principles and the results of a numerical, computerized method of analysis for predicting metal flow and temperature distributions in axisymmetric extrusion, compression and drawing.

In extrusion and drawing, both plastic deformation and friction contribute to heat generation. Approximately 90 to 95 percent of the mechanical energy involved in the process is transformed into heat.[1] For commerical reductions and speeds in extruding and drawing today's materials, temperature increases of several hundred degrees may be involved. A part of the generated heat remains in the deformed material, another flows into tooling while still an additional part may flow into the undeformed portion of the material. The temperatures developed in the process influence the lubrication conditions, the tool life, the properties of the final product, and most significantly, they determine the maximum deformation speed which can be used for producing sound products without excessive tool damage. Thus, temperatures greatly influence the productivity of extrusion, compression (or upsetting) and drawing processes.

During extrusion, compression and drawing, heat is generated by deformation in the material and by friction at the tool-material interface. Heat is transported with the deformed material and conduction takes place simultaneously. Some of the generated heat remains in the product, some is transmitted to the tooling, and some may even increase the temperature of the material coming into the deformation zone.

The general problem to be examined is that of time-dependent heat flow in an incompressible moving medium with heat generation in the medium. In plane strain deformation with two dimensional heat flow, the governing equation is:

$$\alpha \left(\frac{\partial^2 \Theta}{\partial x^2} + \frac{\partial^2 \Theta}{\partial y^2} \right) - \left(u \frac{\partial \Theta}{\partial x} + v \frac{\partial \Theta}{\partial y} \right) + \frac{\beta \dot{W}}{J \rho c} = \frac{\partial \Theta}{\partial t} \qquad (1)$$

where

β = fraction of deformation energy transformed into heat

α = thermal diffusivity (constant)

Θ = temperature rise

x,y = coordinates in x and y directions, respectively

u,v = velocities in x and y directions, respectively

$\dot{W}$ = rate of energy dissipation

J = mechanical equivalent of heat

ρ = specific gravity

c = heat capacity

t = time

The problem, described by Equation 1, involves the determination of simultaneous heat generation, transportation, and conduction. It is impossible to solve this complex problem analytically. Therefore, a numerical method of solution has been developed. This method, originally suggested by Bishop[2], approximates the heat generation and the simultaneous heat conduction in two steps which take place consecutively during equal time increments, Δt. The repetition of these two steps simulates numerically the deformation process and gives the temperature distribution as a function of time.

2. THEORETICAL DETERMINATION OF THE VELOCITY FIELD IN CYLINDER UPSETTING. For problems in metal forming, there are no exact solutions that can be used for practical purposes. Therefore, methods of analysis giving results with various degrees of approximations must be used. Among various theoretical methods available for analyzing metalforming problems, the upper-bound method is the most practical technique for theoretical analysis of metal flow. For describing the metal flow, this method considers an admissible velocity field that satisfies the incompressibility, continuity, and the velocity boundary conditions. Based on this velocity field, the deformation, the shear (if velocity discontinuities are present), and the friction energies are computed to give the total forming energy and also the forming load. Based on limit theorems, this calculated forming load is necessarily higher than the actual load and it, therefore, represents an upper bound to the actual forming load. Thus, the lower this upper-bound load, the better is the prediction. Often the velocity field considered includes one or more parameters that are determined by minimizing the total forming energy with respect to those parameters. Thus, the determined values of the parameters give a somewhat better upper-bound velocity field. In general, with an increasing number of parameters in the velocity field, the solution improves while the computations become more complex. Consequently, for practical use of the upper-bound method, practical compromises are made in selecting an admissible velocity field.

When applying the upper-bound method, the following assumptions are usually made:

1. The deforming material is isotropic and incompressible.

2. The elastic deformations are neglected.

3. The inertial forces are small and neglected (i.e., high-energy-rate forming is not considered).

4. The friction shear stress, τ , is constant at the die-material interface and is related to a constant shear factor, f, or to a friction factor, m, whose definitions are

$$\tau = f\,\bar{\sigma} = m\,\bar{\sigma}/\sqrt{3} \tag{2}$$

where $\bar{\sigma}$ is the flow stress of the material.

5. The material flows according to von Mises' flow rule.

6. The flow stress, $\bar{\sigma}$, is constant.

In upsetting (or compressing) a cylinder symmetric about the z axis, for example, the velocity components in cylindrical coordinates (r,z) can be expressed in terms of an unknown parameter β_1 [3]

$$v = 2Az\left(1 - \frac{\beta_1}{3} z^2\right) \tag{3}$$

where v is the axial velocity component, the constant A is

$$A = V_0/2H\left(1 - \frac{\beta_1}{12} H^2\right) \tag{4}$$

and H is the height of the cylinder.

$$u = A(1-\beta_1 z^2)r \tag{5}$$

$$\dot{\bar{\varepsilon}} = \frac{2A}{\sqrt{3}}\left[\,3\,(1-\beta_1 z^2)^2 + (\beta_1 rz)^2\,\right]^{1/2} \tag{6}$$

Also where u is the radial velocity component, and $\dot{\bar{\varepsilon}}$ the effective rate of strain.

An expression for energy rate is now formed.[3] The parameter β_1 is then found from the condition of minimizing the energy rate with respect to β_1 .

As an example of results, predicted and measured deformations of upset (or compressed) aluminum samples are shown in Figure 1 for two different friction conditions.

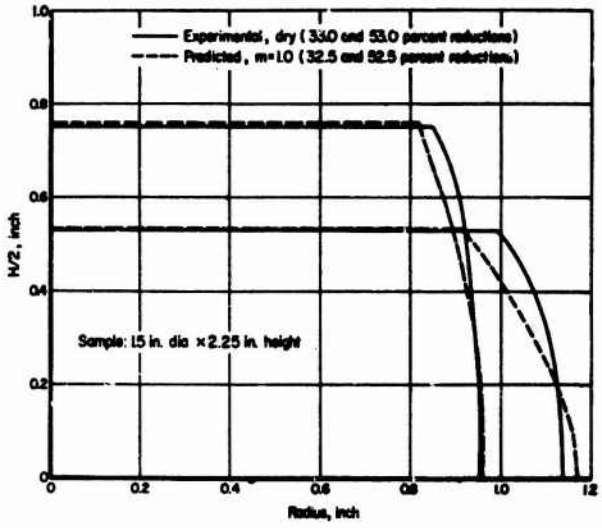


Fig. 1 Bulge profile obtained in dry upsetting of annealed 1100 aluminum cylinders

3. VELOCITIES AND TEMPERATURES IN EXTRUSION AND DRAWING. For estimating a realistic velocity field in extrusion and drawing, Lambert and Kobayashi[4] introduced a method for obtaining upper-bound velocity fields without discontinuities. In axisymmetric flow, this method uses the flow function and the superposition of an infinite number of flow patterns. The calculations are performed numerically using a digital computer. The original method as suggested by Lambert and Kobayashi[4] required considerable amount of computer time for execution. Therefore, in the present study, a simplified version has been derived and programmed in Fortran IV as a system of subprograms called EXTVEL. The program EXTVEL calculates the axial and radial velocities, the strain-rate, and the strain, in the deformation zone, at meshpoints of the grid system as illustrated in Figures 2 and 3. Figures 2 and 3 are taken from Reference 5.

As an example, metal flow in drawing and heat transfer between a volume element and its vicinity are illustrated in Figure 2. The volume element moves by following the flow line and deformation takes place under the die in the deformation zone. In wire drawing, friction takes place only at die-material interface while in extrusion, friction or internal shear occurs at the die as well as at the container surface.

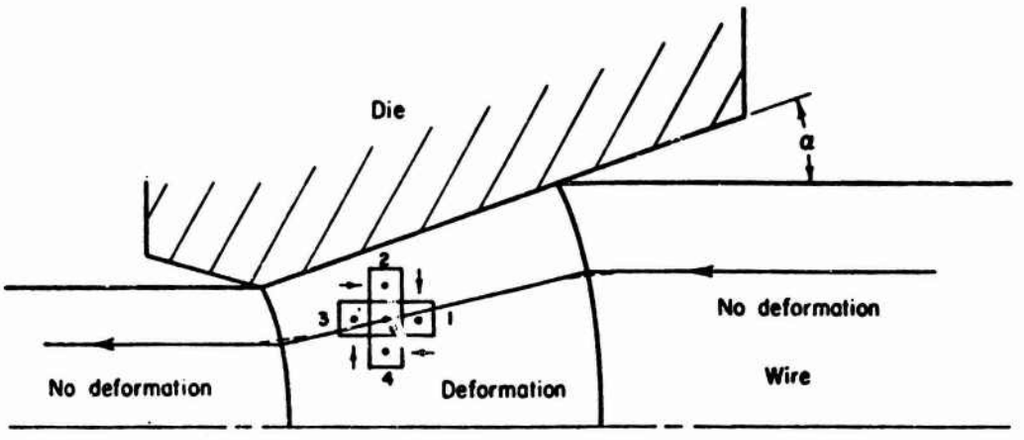


FIGURE 2. Metal Flow and Configuration of Volume Elements Used in Numerical Prediction of Temperatures in Wire Drawing.

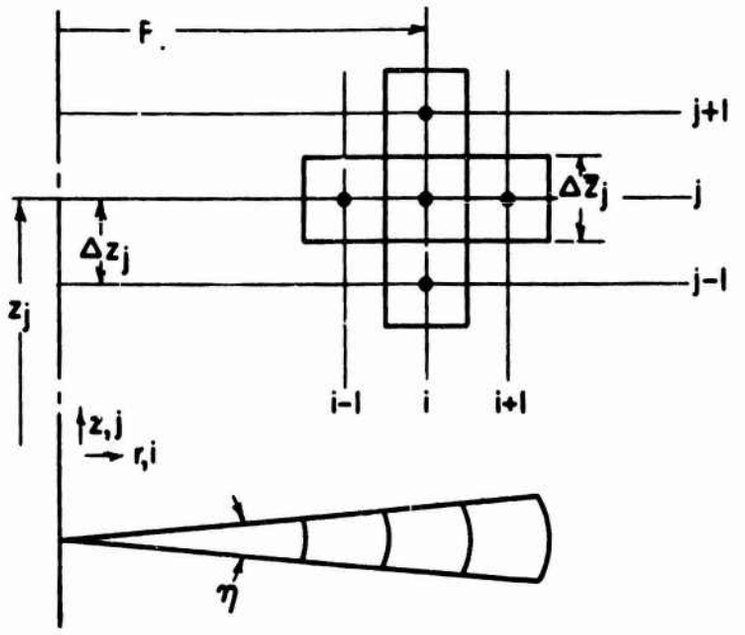


FIGURE 3. Representation of Cylindrical Grid System and Volume Elements for Deriving the Difference Equations of Heat Conduction

For a volume element, the temperature increase due to deformation, θ_d, in a time interval, Δt, is given by:

$$\theta_d = \bar{\sigma} \dot{\bar{\varepsilon}} \Delta t \beta_2 / J\rho c \tag{7}$$

where

θ_d = temperature increase due to deformation

Δt = time interval

c = specific heat of the drawn material

ρ = specific weight of the drawn material

β_2 = fraction of deformation energy transformed into heat.

The temperature increase due to friction is given by:

$$\theta_f = f \bar{\sigma} v F \Delta t / J c_a \rho_a V_a \tag{8}$$

where, in addition to the symbols already described,

θ_f = temperature increase due to friction

f = friction factor at the interface

v = velocity at the interface

F = surface area at the interface

V = volume element

c_a, ρ_a = average specific heat and specific weight at the interface.

During a time interval, Δt, conduction heat transfer takes place between the volume element "0", seen in Figure 2, and between the adjacent volume elements "1", "2", "3", and "4". The temperature change in volume element "0" after conducting during the time element, Δt, is calculated by solving the difference equations of heat transfer. The maximum value of Δt is determined from a stability criterion.

Figure 3 illustrates the notation used in deriving the finite difference equation of heat transfer in its general form. Because of axial symmetry the volume element "i,j" is considered over a rotational angle $\eta = 1$ radian. Considering heat flow in only the axial and radial

directions, a heat balance is found by equating the sum of the amounts of heat conducted from adjacent elements into element "i,j" to the heat stored in this element:

$$\alpha_{j-i} k_{i,j-1} \Delta t\, S \frac{(\theta_{i,j-1} - \theta i,j)}{\Delta z_j} + \alpha_{j+1} k_{i,j+1} \Delta t\, S \frac{(\theta_{i,j+1} - \theta_{i,j})}{\Delta z_{j+1}} +$$

$$+ \alpha_{i+1} K_{i+1,j}\, \Delta \bar{z}_j \Delta t\, (\theta_{i+1,j} - \theta_{i,j}) \frac{\bar{R}_{i+1}}{\Delta R_{i+1}} + \alpha_{i-1} k_{i-1,j} \Delta \bar{z}_j \Delta t\, (\theta_{i-1,j} - \theta_{i,j}) \frac{\bar{R}_i}{\Delta R_i} +$$

$$+ \beta_i h_c\, (\theta_o - \theta_{i,j}) R_i \Delta \bar{z}_j \Delta t + \beta_j h_f\, (\theta_o - \theta_{i,j})\, S\, \Delta t =$$

$$\gamma\, S\, \Delta \bar{z}_j \rho c\, (\theta'_{i,j} - \theta_{i,j}) \,. \qquad (9)$$

where, in addition to the dimensions given in Figure 3,

$K_{i+1,j}$ = heat conductivity between the elements, (i,j) and (i+1,j)

β_i, β_j = factor indicating the portion of the element (i,j) subject to convection and radiation heat transfer in radial and axial directions, respectively.

$\theta_{i,j}, \theta'_{i,j}$ = temperatures of element (i,j) before and after time element Δt, respectively.

Δt = time element during which heat transfer takes place

S = surface area of element (i,j) in axial projection

h_c, h_f = heat transfer coefficient (convection and radiation) at the free surfaces of a cylinder in radial and axial directions, respectively

θ_o = temperature of environment

γ = portion of the element (i,j) subject to temperature variation

α_i, α_j = factor indicating the portion of the element (i,j) subject to heat conduction in radial and axial directions, respectively

ρ, C = specific gravity and specific heat of volume element (i,j), respectively.

Equation 9 is in very general form and can be used for all boundary conditions which occur in the present problem. According to each boundary condition, the thermal constants and the factors, α , β , and γ must be modified. For example, at the interior of the product, $\alpha_i=\alpha_j=1$, $\beta_i=\beta_j=0$, $\gamma=1$ and all $k_{i,j}$'s are equal to the heat conductivity of billet material at the temperature $\theta_{i,j}$. Similarily, at the free cylindrical surface of the product, $\alpha_{j-1} = \alpha_{+1} = 0.5$, $\alpha_{i-1}=1$, $\alpha_{i+1}=0$, $\beta_i=1$, $\beta_j=0$, $\gamma=0.5$.

Using the above numerical procedures the extrusion of an aluminum billet (rod) through a flat die was simulated. Figure 4 shows the isotherms when the ram displacement is 3.7 inches.

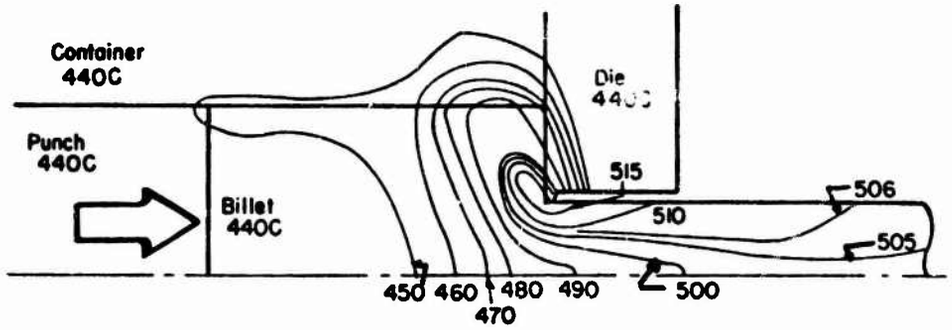


FIGURE 4. Temperature Distributions in Extrusion of Al 5052 Alloy Rod Through a Flat Die

Reduction = 5:1, Ram Speed = 74.4 in/min, Billet Diameter = 2.8 in, Billet Length - 5.6 in, Initial Billet and Tooling Temperatures = 440 C, Ram Displacement = 3.7 inch.

4. CONCLUSIONS AND FUTURE WORK

1. Based on the above and related studies, it is concluded that the various bounding theorems of plasticity, when combined with numerical heat conduction/convection, provide an excellent approach to predicting practical metalforming characteristics.

2. Continued future efforts will include the modeling of drawing and nosing of hot shell and cartridge cases.

REFERENCES

1. Farren, W. S. and Taylor, G. I., "The Heat Developed During Plastic Extrusion of Metals", Proceedings of the Royal Society, Series A, 1925, Vol 107, p. 422.

2. Bishop, J. F. W., "An Approximate Method for Determining the Temperatures Reached in Steady State Motion Problems of Plane Plastic Strain", Quarterly J. of Mech. and Appl. Math., Vol. 9, 1956, p. 236.

3. Lee, C. H. and Altan, T., "Influence of Flow Stress and Friction Upon Metal Flow in Upset Forging Rings and Cylinders", J. Engr. for Industry, Aug 1972, p. 775-779.

4. Lambert, E. R. and Kobayashi, S., "Admissible Velocity Fields for Some Steady-State Forming Processes in Plane Strain and Axisymmetry", Presented at the TSME, 1967, Semi-International Symposium, Tokyo, Septemter 1967.

5. Altan, T., et. al., "Approximate Calculation of Velocity and Temperature Distributions in Axisymmetric Extrusion and Drawing", Proc. North Amer. Metalworking Res. Conf., McMaster Univ., Hamilton, Ontario, Canada, 1973, Vol 1, pp 107-127.

CONJUGATE DIRECTION METHODS IN OPTIMIZATION

Magnus R. Hestenes
Professor of Mathematics
University of California at Los Angeles
Los Angeles, California 90024
and Visiting Research Mathematician
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

ABSTRACT. The present paper is concerned with iterative methods for finding the minimum of a function of n variables. We begin by a discussion of the gradient method and of Newton's method. It is pointed out that effective methods for minimizing a quadratic function can be extended so as to obtain the minimum of a nonquadratic function. Accordingly several algorithms for minimizing a quadratic function are given together with their extensions. These algorithms are based upon a concept of conjugacy and are called methods of conjugate directions. In particular we discuss the conjugate gradient algorithm, the method of parallel planes, the method of parallel displacements and the conjugate Gram Schmidt process. We conclude with a description of matrix forms of these algorithms.

1. INTRODUCTION. One of the basic problems in optimization is the determination of efficient algorithms for finding the minimum of a function $f(x)$ on a set S. In the present paper we shall be concerned with the unconstrained case, that is, to the case in which the minimum point x_o of f is an interior point of S. Algorithms of this type also can be used for solving constrained minimum problems. This is because for a large class of constrained problems the minimum point is also the solution of an equivalent unconstrained minimum problem. In general these equivalent unconstrained problems are the limits of sequences of easily constructed unconstrained minimum problems.

2. PRELIMINARY REMARKS. In the present paper we shall be concerned mainly with the problem of minimizing a function $f(x)$ of n real variables $x = (x^1, \ldots, x^n)$ on an open set S. This is the so-called unconstrained minimum problem.

Much of our analysis will be based on the concept of directional derivatives. Given a point x and a vector $p \neq 0$ the derivatives of the function

$$\varphi(\alpha) = f(x+\alpha p)$$

with respect to α at $\alpha = 0$ are called the directional derivatives of f in the direction p. In particular the derivative

$$\varphi'(0) = f'(x, p) = \sum_{i=1}^{n} \frac{\partial f(x)}{\partial x^i} p^i \tag{1}$$

is the directional derivative of f in the direction p of the first order. The second order directional derivative is given by the formula

$$\varphi''(0) = f''(x, p) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 f(x)}{\partial x^i \partial x^j} p^i p^j \quad . \tag{2}$$

The second order Taylor's formula for f takes the form

$$f(x+p) = f(x) + f'(x, p) + \tfrac{1}{2} f''(x, p) + R_2(x, p) \tag{3}$$

where $R_2(x, p)$ is the remainder. For the functions with which we will be concerned we have

$$\lim_{p \to 0} \frac{R_2(x, p)}{|p|^2} = 0$$

Here $|p|$ is the length of p.

The vector $\left(\frac{\partial f(x)}{\partial x^i}\right)$ is called the <u>gradient</u> of f at x and will be denoted by $\nabla f(x)$ or by $f'(x)$. It is in the direction of steepest ascent of f. The matrix

$$f''(x) = \left(\frac{\partial^2 f(x)}{\partial x^i \partial x^j}\right)$$

is called the Hessian of f at x and will be denoted by $f''(x)$. In vector and matrix notations we have

$$f'(x, p) = p^* \nabla f(x) = \nabla f(x)^* p, \quad f''(x, p) = p^* f''(x) p \tag{4}$$

where p and ∇f are column vectors and $p^*, \nabla f^*$ are the corresponding row vectors. In this notation p^*q denotes the innerproduct of two vectors p and q. By a quadratic function $F(x)$ will be meant one that is expressible in the form

$$F(x) = \tfrac{1}{2} x^* A x - k^* x + \text{constant} \tag{5}$$

where A is a symmetric constant matrix and k is a constant vector. Its gradient is

$$\nabla F(x) = F'(x) = Ax - k \quad . \tag{6}$$

We shall restrict ourselves to quadratic functions for which A is a positive definite matrix. In this event the level surfaces F(x) = constant are similar ellipsoids whose common center x_o is the minimum point of F. At a minimum point x_o of a general function f we have

$$\nabla f(x_o) = 0 \quad , \quad f''(x_o, p) \geqq 0$$

for all $p \neq 0$. If det $f''(x_o) \neq 0$, as we shall assume, we have

$$f''(x_o, p) = p^* f''(x_o)\, p > 0$$

unless $p = 0$. In the neighborhood of x_o the level surfaces of f are ellipsoidal in character, as indicated schematically in Figure 1. At a point

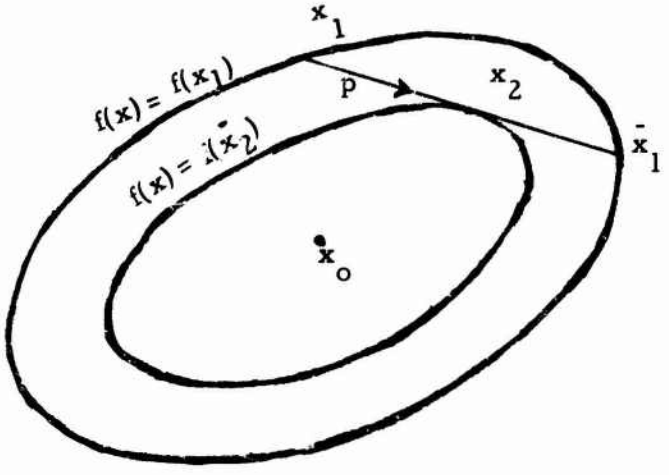


Figure 1

$x = x_1$ select a direction p such that $f'(x_1, p) < 0$. Then p is a direction of descent. At the minimum point x_2 of f on the line $x = x_1 + \alpha p$ the gradient $\nabla f(x_2)$ is orthogonal to this line. It follows that this line is tangent to the level surface $f(x) = f(x_2)$ at x_2. If f is quadratic the point x_2 is the midpoint of the chord $x_1\bar{x}_1$ shown in the diagram. If f is nearly quadratic the point x_2 will be near the midpoint of this chord.

In some instances one must be very close to the minimum point before the level surfaces are elliptical in character. For example in the two dimensional case the level surface of the function

$$f(x, y) = (x-1)^2 + 100(y - x^2)^2$$

is shape more like a banana than an ellipse and is accordingly called a banana function. This function is used to illustrate difficulties that may arise in finding the minimum of a function by iterative procedur. Its

minimum point is obviously at the point (1, 1).

In this paper we shall be concerned with linear iterations of the form

$$x_{k+1} = x_k + a_k p_k . \tag{7}$$

Here p_k is a direction of correction and a_k is a scalar. We normally choose p_k in a direction of descent, that is, a direction p_k such that $f'(x_k, p_k) < 0$. If $f'(x_k, p_k) > 0$ we can replace p_k by $-p_k$ to obtain a direction of descent. This is equivalent to changing the sign of a_k. For directions of descent the scalar a_k is chosen to be positive. If $f'(x_k, p_k) = 0$ we set $a_k = 0$ and proceed to the next step. This is equivalent to disregarding the direction p_k. If $f'(x_k, p) = 0$ for all p, then $\nabla f'(x_k) = 0$ and x_k is a critical point of f. It is a local minimum point if $f''(x_k, p) > 0$ for all $p \neq 0$.

Perhaps the most obvious choice of p_k is the direction $p_k = -\nabla f(x_k)$ of steepest descent. An iteration in which this choice is made at each step is called a gradient method. As we shall see later, we can modify the gradient method in various ways so as to obtain more efficient algorithms.

Having chosen the vector p_k various rules can be given for choosing the scalar a_k in the iteration (7). Typical choices are the following

(i) Choose $a = a_k$ so as to minimize f along the line $x = x_k + a p_k$. This involves some type of search routine unless a formula for a_k can be given as in the quadratic case. This value of a is called the optimal a. Fortunately in most instances all that is needed is a rough estimate of the optimal a.

(ii) Guess the value of a. If $f(x_k + a p_k) > f(x_k + 2a p_k)$ choose $a_k = 2a$. If on the other hand $f(x_k + \frac{a}{2} p_k) < f(x_k + a p_k)$ choose $a_k = a/2$. Otherwise set $a_k = a$. A judicious program of this type is usually effective.

(iii) Choose $a_k = c_k/d_k$, where $c_k = f'(x_k, p_k)$ and $d_k = f''(x_k, p_k)$. This represents one linear Newton step for $f(x_k + a p_k)$ as a function of a. In the quadratic case this choice of a_k is optimal. In order to avoid computation of second derivatives one may use the difference formula

$$d_k = \frac{f'(x_k + \sigma_k p_k, p_k) - f'(x_k, p_k)}{\sigma_k}$$

for d_k. If one restricts one-self to evaluations of functional values only

one can use the central differences.

$$c_k = \frac{f(x_k - \sigma_k p_k) - f(x_k + \sigma_k p_k)}{2\sigma_k}, \quad d_k = \frac{f(x_k - \sigma_k p_k) - 2f(x_k) + f(x_k + \sigma_k p_k)}{\sigma_k^2}.$$

Here $\sigma_k = \frac{\sigma}{|p_k|}$, where σ is a small constant. Near the minimum point σ_k should be of the same order of magnitude as a_k.

3. GRADIENT METHODS. Recall the a solution the differential equations

$$\frac{dx}{dt} = -\nabla f(x)$$

is a path of steepest descent. It normally terminates at the minimum point of f. If we use the Euler method for estimating a solution to this equation we obtain ie iteration

$$x_{k+1} = x_k + \Delta x_k, \quad \Delta x_k = -\Delta t_k \nabla f(x_k).$$

This iteration is the form

$$x_{k+1} = x_k - a_k \nabla f(x_k) \tag{8}$$

with $a_k = \Delta t_k$. This algorithm is called the gradient algorithm. This derivation of the gradient algorithm suggests that we can select a_k = constant in the iteration (8). This choice is of the type (ii) given in the last section. On the other hand we may choose a_k optimally in the sense that $\alpha = a_k$ minimizes the function f on the line $x = x_k - \alpha \nabla f(x_k)$. This choice is excellent if the level surfaces of f are nearly spherical. However if its level surfaces are elongated ellipsoids convergence will be very slow. This fact is illustrated by the following two dimensional example. in which we minimize the function

$$f(x, y) = \tfrac{1}{2}(x^2 + \gamma y^2) \qquad (\gamma = .001).$$

Starting with $(x_1, y_1) = (1, \gamma)$ we choose $a_k = \beta \alpha_k$, where α_k is optimal. This yields the iteration

$$x_{k+1} = x_k - \beta \alpha_k x_k, \quad y_{k+1} = y_k - \beta \alpha_k \gamma y_k$$

where α_k minimizes f on the line $x = x_k - \alpha x_k$, $y = y_k - \alpha \gamma y_k$.

A formula for α_k is $\alpha_k = \dfrac{x_k^2 + \gamma y_k^2}{x_k^2 + \gamma^2 y_k^2}$ The number N of iterations

needed to obtain $|x_{k+1}| < 10^{-6}$, $|y_{k+1}| < 10^{-6}$ is given in the following table.

β	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
N	2272	1843	1227	862	504	565	435	357	246	6908

This table shows that for this example the optimal choice $\beta = 1$ is the poorest of the values of β considered. One should not conclude from this example that $\alpha = .9$ is the best choice. Examples can be constructed for which the choice $\beta = .7$ is preferable to $\beta = .9$. If conjugate gradients are used the solution can be obtained in two steps.

The concept of the gradient is dependent in the metric used. If we change the metric as is the case when we transform coordinates the gradient is altered. Analytically the gradient of f is a vector g such that the directional derivate $f'(x, h)$ of f at x is expressible in the form

$$f'(x, h) = \langle g, h \rangle$$

where $\langle x, y \rangle$ is the inner product. If we choose as our inner product

$$\langle x, y \rangle = x^* H^{-1} y ,$$

where H is a positive definite symmetric matrix, then

$$f'(x, h) = \nabla f(x)^* h = g^* H^{-1} h$$

for all vectors k. In this event $H^{-1} g = \nabla f(x)$ and $g = H \nabla f(x)$ is our gradient. It follows that the algorithm

$$x_{k+i} = x_k - a_k H \nabla f(x) \tag{9}$$

is also a gradient algorithm. In order to see a connection between the algorithms (8) and (9) select a matrix M such that $H = MM^*$. Under the transformation $x = My$ the function $f(x)$ becomes $F(y) = f(My)$. The gradient of F with respect to y is

$$\nabla F(y) = M^* \nabla f(My) .$$

Consequently the standard gradient algorithm for y is

$$y_{k+1} = y_k - a_k M^* \nabla f(My_k).$$

Inasmuch as $x = My$ we have

$$x_{k+1} = My_{k+1} = My_k - a_k MM^* \nabla f(My_k) = x_k - a_k H \nabla f(x_k).$$

It follows that the generalized gradient algorithm (9) is the usual gradient algorithm in another coordinate system. If we select H to be the inverse of the Hessian $f''(x_o)$ at the minimum point x_o, the level surfaces for $F(y) = f(My)$ with $MM^* = f''(x_o)^{-1}$ become spherical in form near $y_o = Mx_o$ and the algorithm

$$x_{k+1} = x_k - a_k f''(x_o)^{-1} \nabla f(x)$$

with $a_k = 1$ converges rapidly. In fact it is normally quadratic. Since x_o is unknown this algorithm cannot be used. However as an alternate procedure we can select a new H at each step giving us an algorithm of the form

(10) $$x_{k+1} = x_k - a_k H_k \nabla f(x_k) .$$

The case $H_k = f''(x_k)^{-1}$, $a_k = 1$ is Newton's algorithm as will be seen in the next section.

4. NEWTON's ALGORITHM. In discussing Newton's method it will be convenient at times to use the alternative symbol $f'(x)$ for $\nabla f(x)$. At a minimum point x_o of f we have $f'(x_o) = 0$. The problem at hand is therefore to find a solution of $f'(x) = 0$. If x_k is an estimate of x_o our problem is therefore to solve for h in the equation

$$f'(x_k + h) = f'(x_k) + f''(x_k)h + r_k(x, h) = 0 .$$

This cannot be done exactly. If we are near the solution the remainder term r_k is small and we can obtain a good estimate h_k by disregarding r_k and solving the equation

(11) $$f'(x_k) + f''(x_k)\, h = 0$$

The solution is $h_k = -f''(x_k)^{-1} f'(x_k)$. This yields the iteration

(12) $$x_{k+1} = x_k - f''(x_k)^{-1} f'(x_k) = x_k - f''(x_k)^{-1} \nabla f(x_k) .$$

This algorithm is known as Newton's algorithm. It is of the form (10) with $a_k = 1$ and $H_k = f''(x_k)^{-1}$ and is accordingly a generalized gradient method. It can be shown to have superlinear convergence if f is of class C'' and quadratic convergence if f is of class C''.

Newton's method can be viewed as a minimization procedure as follows. Given an estimate x_k of our solution consider the Taylor expansion

$$f(x_k + h) = f(x_k) + f'(x_k) * h + \tfrac{1}{2} h^* f''(x_k) h + R_k(h).$$

Disregard the remainder R_k and find the minimum h_k of the quadratic function

$$F_k(h) = f(x_k) + f'(x_k)*h + \tfrac{1}{2}\, h*f''(x_k)\, h \,. \tag{13}$$

At the minimum point the gradient of $F_k(h)$ with respect to h must vanish. Hence h_k is a solution of the equation

$$F_k'(h) = f'(x_k) + f''(x_k)\, h = 0.$$

This equation is identical with equation (11). In view of this result the Newton algorithm (12) can be viewed as follows. Select an initial point x_1. Having obtained an estimate x_k of x_o select the next estimate $x_{k+1} = x_k + h_k$ by choosing h_k to be the minimum of the second order approximation $F_k(h)$ of $f(x_k+h)$ given by equation (13). Consequently Newton's method is determined by a sequence of minimizations of suitably chosen quadratic functions. Accordingly it is appropriate to study effective methods for minimizing quadratic functions. This will be done in the sections that follow.

As remarked above Newtons iteration is of the form

$$x_{k+1} = x_k + h_k \,, \quad f'(x_k) + H_k^{-1} h_k = 0$$

with $H_k^{-1} = f''(x_k)$. It can be shown that superlinear convergence is preserved if we replace H_k by any positive definite matrix such that H_k converges to $f''(x_o)^{-1}$. One choice of H_k is the inverse of the matrix A whose column vectors are

$$\frac{f'(x_k + \sigma_k u_j) - f'(x_k)}{\sigma_k} \qquad (j = 1, \ldots, n)$$

where $u_1, \ldots, u_n$ are linearly independent unit vectors and σ_k converges to zero as k becomes infinite. This yields a version of the secant method for solving the equation $f'(x) = 0$.

5. MINIMIZATION OF QUADRATIC FUNCTIONS. Our discussion of Newton's method suggests that efficient methods for obtaining the minimum of quadratic functions can be modified so as to obtain efficient methods for minimizing a nonquadratic function. In order to apply this principle it is essential to develop methods for minimizing a quadratic function

$$F(x) = \tfrac{1}{2}\, x*Ax - k*x + \text{constant}. \tag{13}$$

Here A is a positive definite symmetric matrix and k is a fixed vector.

The constant term is of no consequence. Inasmuch as the gradient of F is $F'(x) = Ax - k$. The problem of minimizing F is equivalent to the problem of solving the linear equation

$$Ax = k.$$

Of course this can be done by an elimination method. There are various types of elimination methods, most of which can be viewed as special cases of the method of conjugate directions which we shall describe presently.

The level surfaces $F(x) =$ constant form as one parameter family of similar ellipsoids whose common center is the minimum point x_o of F. Accordingly the problem of minimizing F is equivalent to the geometric problem of finding the center of an ellipsoid.

We have the following basic property of positive definite quadratic functions F.

π_{n-1} The minimum points of F on parallel lines lie on an (n-1) plane through the minimum point x_o of F.

This fact is shown schematically in the following diagram

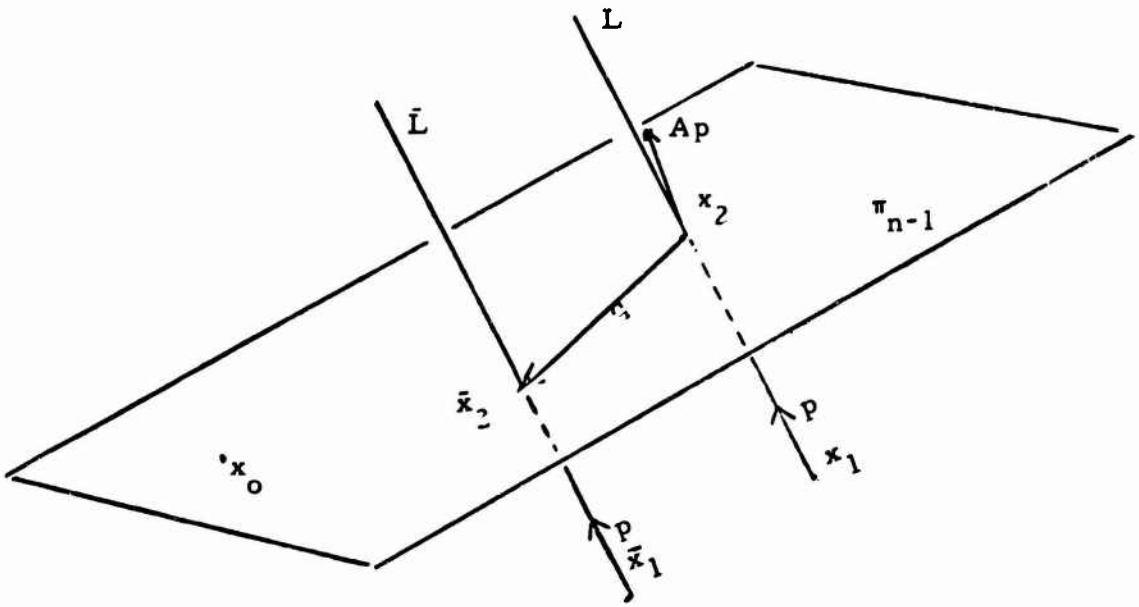


Figure 2

Let L: $x = x_1 + ap$ be a line through x_1 in the direction p. At the minimum point x_2 the gradient $F'(x_2) = Ax_2 - k$ of F must be orthogonal to L and hence to the vector p. The equation

$$p^*(Ax-k) = 0$$

therefore holds when $x = x_2$. This is the equation of an $(n-1)$-plane π_{n-1} through x_2 having Ap is its normal. Since $Ax_o = k$ the minimum point x_o of F lies in π_{n-1}. If $\bar{L}$: $x = \bar{x}_1 + ap$ is a line parallel to L, the minimum point $\bar{x}_2$ of F on $\bar{L}$ also lies in π_{n-1} in view of the fact that $F'(\bar{x}_2) = A\bar{x}_2 - k$ is orthogonal to the direction p of $\bar{L}$. From the equations

$$p^*(Ax_2 - k) = 0, \quad p^*(A\bar{x}_2 - k)$$

it is seen that $p^*A(\bar{x}_2 - x_2) = 0$. It follows that the vector $q = \bar{x}_2 - x_2$ satisfies the equation

$$p^*Aq = 0 \quad . \tag{14}$$

This relation expresses a geometrical phenomenon called "conjugacy". Accordingly we say that two vectors p and q are <u>conjugate</u> if the A-orthogonality relation (14) holds. More generally a k-plane π_k is said to be conjugate to p if Ap is normal to π_k. In particular the $(n-1)$ - plane π_{n-1} shown in Figure 2 is conjugate to p.

A set of vectors $p_1, \ldots, p_n$ are said to be mutually conjugate if

$$p_j^* Ap_k = 0 \ (j \neq k) \ , \quad d_k = p_k^* Ap_k > 0. \tag{15}$$

The fact that $d_k > 0$ excludes the vector $p_k = 0$. It is clear that vectors $p_1, \ldots, p_n$ are linearly independent and hence form a basis for our Euclidean n-space. The vectors $Ap_1, \ldots, Ap_k$ also form a basis called the <u>conjugate basis</u> of the basis $p_1, \ldots, p_n$.

Suppose that we have given a set of mutually conjugate vectors $p_1, \ldots, p_n$. Given a point x_1, the vectors $p_1, \ldots, p_k$ determine a k-plane

$$\pi_k: \quad x = x_1 + a_1 p_1 + \ldots + a_k p_k$$

where $a_1, \ldots, a_k$ are arbitrary parameters. It will be convenient to call the direction

$$r = -F'(x) = k - Ak$$

of steepest descent the <u>residual</u> of F at x. At the minimum point

$$x_{k+1} = x_1 + a_1 p_1 + \ldots + a_k p_k \tag{16}$$

of F on π_k the direction of steepest descent is

(17) $$r_{k+1} = k - Ax_{k+1} = r_1 - a_1 Ap_1 - \ldots - a_k Ap_k.$$

It is perpendicular to π_k and hence to each of the vectors $p_1, \ldots, p_k$. Hence, by (15)

$$0 = p_j^* r_{k+1} = p_j^* r_1 - a_j p_j^* Ap_j.$$

The scalars $a_1, \ldots, a_k$ are given by the formula

(18) $$a_j = c_j/d_j, \quad c_j = p_j^* r_1, \quad d_j = p_j^* Ap_j.$$

Observe further that if we set

$$x_k = x_1 + a_1 p_1 + \cdots + a_{k-1} p_{k-1}$$

the (k-h)-plane

$$x' = x_h + a_h p_h + \cdots + a_\kappa p_\kappa$$

is a subplane of π_k. Hence

$$x_{k+1} = x_h + a_h p_h + \cdots + a_k p_k$$

minimizes F on this subplane it follows that (18) holds with $c_j = p_i^* r_h$ $(j = h, \ldots, p)$. We have accordingly the relations

(19) $$c_j = p_j^* r_h \qquad (h = 1, \ldots, r)$$

Consequently equations (18) are equivalent to the equations

(20) $$a_j = c_j/d_j, \quad c_j = p_j^* r_j, \quad d_j = p_j^* Ap_j.$$

We normally use equations (20) to determine the scalar a_j in equation (16).

We are now in a position to establish the following proposition.

Let mutually conjugate vectors $p_1, \ldots, p_n$ be given. Starting with an initial point x_1 the minimum point of F is obtained by minimizing F successively in the direction $p_1, \ldots, p_n$.

This situation is shown schematically in Figure 3. At the

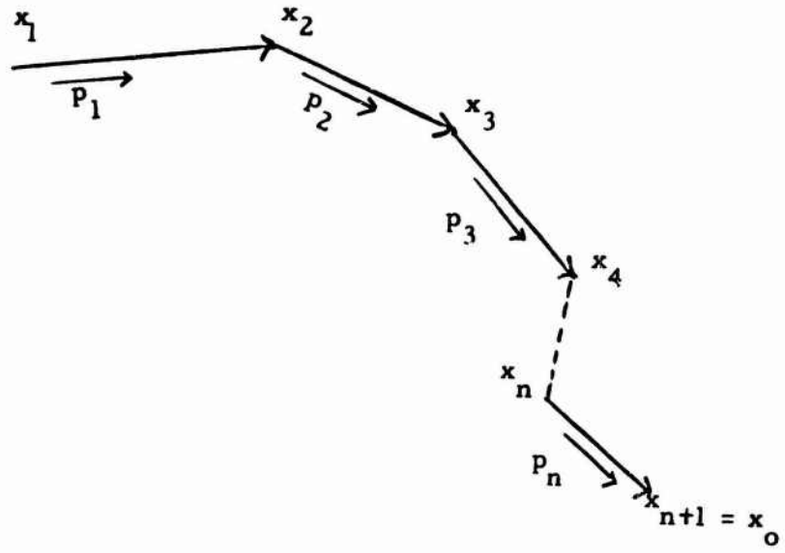


Figure 3

k-th step we minimize F on the line $x = x_k + ap_k$. The minimum point $x_{k+1} = x_k + a_k p_k$ of F on the line is obtained when a_k is given by the formula (20) with j = k. In view of the results given above the point x_{k+1} obtained by these successive minimizations is the minimum point of F on the k-plane π_k. Consequently point x_{n+1} obtained on the n-th step is the minimum point x_o of F on our n-space.

The proposition given above can be formalized as an algorithm as follows

(a) select an initial point x_1 and an initial direction p_1

(b) Given x_k and p_k find the minimum point

$$x_{k+1} = x_k + a_k p_k$$

on the line $x = x_k + a p_k$. Here a_k is given by

$$a_k = c_k/d_k \,, \quad c_k = p_k^* r_k \,, \quad d_k = p_k^* A p_k$$

(c) Select p_{k+1} conjugate to $p_1, \ldots, p_k$.

This algorithm terminates at the m-th step if $r_{m+1} = 0$. We call an algorithm of this type a method of conjugate directions. Various algorithms of this type will be obtained by specifying formulas for computing the conjugate directions $p_1, \ldots, p_n$.

6. METHOD OF CONJUGATE GRADIENTS. Perhaps the simplest conjugate direction method is the method of conjugate gradients. It can be

described as follows: select an initial point x_1 and compute its negative gradient $p_1 = -F'(x_1)$. Find the minimum point x_2 of F on the line $x = x_1 + \alpha p_1$. Let π_{n-1} be the (n-1)-plane through x_2 conjugate to the vector p_1. It contains the minimum point x_o of F. Next compute the negative gradient p_2 of F at x_2 in the conjugate (n-1)-plane π_{n-1}. Presently a formula for p_2 will be given. For the moment it suffices to describe p_2 as determining the direction of steepest descent of F at x_2 in π_{n-1}. We now minimize F along the line $x = x_2 + \alpha p_2$ to obtain a next estimate x_3 of our solution. The (n-2)-plane π_{n-2} through x_3 and conjugate to p_1 and p_2 contains the minimum point x_o of F. Repeat this process. Having obtained the point x_k select the direction p_k of steepest descent of F at x_k in the (n-k+1)-plane π_{n-k+1} through x_k conjugate to $p_1, \ldots, p_{k-1}$. The next estimate x_{k+1} minimizes F on the line $x = x_k + \alpha p_k$. Since each of these planes contain the minimum point x_o of F and their dimensionality is decreased by one at each step the solution x_o is obtained in at most n steps.

The algorithm just stated is somewhat involved. Fortunately, in its application one doest not need to determine the planes $\pi_{n-1}, \pi_{n-2}, \cdots$ explicitly. All that is needed are the recursion formulas

$$p_1 = -F'(x_1), \quad p_{k+1} = -F'(x_{k+1}) + \frac{|F'(x_{k+1})|^2}{|F'(x_k)|^2} p_k$$

for the negative conjugate gradients of F. The justification for these formulas can be found in the references given below.

According to these remarks the conjugate gradient algorithm (cg-algorithm) can be put in the following form: select an initial point x_1 and set $p_1 = -F'(x_1)$. For $k = 1, 2, 3 \ldots$ perform the iteration

(i) Find the minimum point x_{k+1} of F on the line $x = x_k + \alpha p_k$.

(ii) Set $p_{k+1} = -F'(x_{k+1}) + \frac{|F'(x_{k+1})|^2}{|F'(x_k)|^2} p_k$.

If no roundoff errors are encountered the solution is obtained in $m \leq n$ steps. In fact the number of steps needed cannot exceed the number of distinct eigenvalues of the Hessian $F''(x) = A$ of F. If the eigenvalues of A are clustered good estimates of the minimum are obtained in a number of steps equal to the number of clusters.

Since F is quadratic the minimum of $F(x_k + \alpha p_k)$ is given by the formula $\alpha = a_k$, where

$$a_k = -\frac{F'(x_k, p_k)}{F''(x_k, p_k)} = \frac{p_k^* r_k}{p_k^* A p_k}, \quad r_k = -F'(x_k).$$

Using this formula the cg-algorithm (21) is defined by the following formulas

(22a) x_1 arbitrary, $r_1 = -F'(x_1)$, $p_1 = r_1$, $s_1 = Ap_1$

(22b) $x_{k+1} = x_k + a_k p_k$, $r_{k+1} = r_k - a_k s_k$, $p_{k+1} = r_{k+1} + b_k p_k$

(22c) $s_k = Ap_k$, $c_k = p_k^* r_k$, $d_k = p_k^* s_k$, $a_k = c_k/d_k$, $b_k = \dfrac{|r_{k+1}|^2}{|r_k|^2}$.

These and other equivalent formulas were given by Hestenes and Stiefel.

The cg-algorithm (21) is applicable at once to the nonquadratic case if one introduces a linear search routine for minimizing $F(x_k + ap_k)$. This variation was introduced by Fletcher and Reeves and accordingly is known as the Fletcher-Reeves algorithm. After n-steps the algorithm is to be restarted. One can also modify the algorithm (22) so as to be applicable to nonquadratic functions without the computations of second derivatives. This can be done by using the difference formula

(23) $$s_k = \frac{F'(x_1 + \sigma_k p_k) - F'(x_1)}{\sigma_k}, \quad \sigma_k = \sigma/|p_k|$$

for s_k in place of the formula $s_k = Ap_k$. Here σ is a small positive constant. The value $\sigma = 10^{-5}$ has been used effectively by the author on standard test problems.

There is an alternative version of the cg-algorithm which we shall call the method of alternate minimization. In this version we introduce the auxiliary function

$$\hat{F}(x) = \tfrac{1}{2}|F'(x)|^2.$$

Clearly $\hat{F}(x)$ has the same minimum point as F starting with an initial point x_1. Set $\hat{x}_1 = x_1$ and $p_1 = -F'(x_1)$. We find the minimum points x_2 and $\hat{x}_2$ of F and $\hat{F}$ on the line $x = x_1 + ap_1$. Setting $p_2 = -F'(\hat{x}_2)$ we minimize F on the line $x = x_2 + a p_2$ to obtain x_3. We next minimize $\hat{F}$ on the line joining $\hat{x}_2$ to x_3 to obtain $\hat{x}_3$. Setting $p_3 = -F'(\hat{x}_3)$ we find the minimum point x_4 of F on the line $x = x_3 + a p_3$ and determine the minimum point $\hat{x}_4$ on the line joining $\hat{x}_3$ to x_4. Continuing this procedure of alternate minimizations of F on $x = x_k + a p_k$ with $p_k = -F'(x_k)$ and of $\hat{F}$ on the line joining $\hat{x}_k$ to x_{k+1} we reach a

situation in which $\hat{x}_{k+1} = x_{k+1}$. Then the point x_{k+1} is the common minimum point of F and $\hat{F}$.

The extension of alternate minimizations to nonquadratic functions is immediate.

7. METHOD OF PARALLEL PLANES. The conjugate gradient method can be viewed as a special case of a general method which we shall call the method of parallel planes. It can be described as follows. Select an initial point x_1 and find the minimum point x_2 of F on a line π_1. Next obtain the minimum point $\bar{x}_2$ of F on a line $\bar{\pi}_1$ parallel to and distinct from π_1. Then minimize F on the line through the point x_2 and $\bar{x}_2$. The minimum point x_3 on this line minimizes F on the 2-plane π_2 spanning π_1 and $\bar{\pi}_1$. We now proceed to find the minimum point $\bar{x}_3$ on a 2-plane $\bar{\pi}_2$. We then determine the minimum point x_4 of F on the 3-plane π_3 spanning π_2 and $\bar{\pi}_2$ by minimizing F on the line through x_2 and $\bar{x}_2$. Proceeding in this manner we obtain the points $x_2, x_3, \ldots, x_{n+1}$ on successible planes $\pi_1, \pi_2, \ldots, \pi_n$. Since π_n is the whole space the minimum point x_{n+1} of F on π_n is the desired minimum point of F.

The method of parallel planes can be formalized as follows:

(i) Select an initial point x_1 and an initial direction $p_1 = u_1$. Find the minimum point

(24a) $$x_2 = x_1 + a_1 p_1$$

of F on the 1-plane π_1: $x = x_1 + a_1 p_1$.

(ii) Having obtained the minimum point

(24b) $$x_k = x_1 + a_1 p_1 + \ldots + a_{k-1} p_{k-1}$$

of F on the $(k-1)$- plane

(24c) $$\pi_{k-1}: \; x = x_1 + a_1 p_1 + \ldots + a_{k-1} p_{k-1}$$

choose a vector u_k not in π_{k-1} and a point $\hat{x}_k$ in π_{k-1}. On the $(k-1)$-plane

(24d) $$\bar{\pi}_{k-1}: \; x = \hat{x}_k + u_k + a_1 p_1 + \ldots + a_{k-1} p_{k-1}$$

parallel to π_{k-1} find the minimum point

(24e) $$\bar{x}_k = \hat{x}_k + u_k + a_{k1}p_1 + \cdots + a_{k,k-1}p_{k-1}$$

of F and set

(24f) $$p_k = \bar{x}_k - x_k = u_k + \sum_{j=1}^{k-1} (a_{kj} - a_j)p_j + \hat{x}_k - x_1.$$

Finally obtain the minimum point

(24g) $$x_{k+1} = x_k + a_k p_k = x_1 + a_1 p_1 + \cdots + a_k p_k$$

of F on the line $x = x_k + a_k p_k$ and hence on the k-plane

$$\pi_k : x = x_1 + a_1 p_1 + \cdots + a_k p_k.$$

The vectors $p_1, \ldots, p_n$ generated by this algorithm are mutually conjugate. The conjugacy of $p_k = \bar{x}_k - x_k$ to $p_j (j<k)$ follows from the fact that x_k and $\bar{x}_k$ respectively minimize F on the parallel lines $x = x_k + \alpha p_j$ and $x = \bar{x}_k + \alpha p_j$. In as much as the vectors $p_1, \ldots, p_{k-1}$ are mutually conjugate the minimum point $\bar{x}_k$ of F on π_{k-1} can be obtained by a cd-process with $x_{k1} = \hat{x}_k + u_k$ as the initial point. In the event $\bar{x}_k = x_{kk}$, where $x_{k1}, \ldots, x_{kk}$ are generated by the cd-algorithm

(2.5) $x_{k,j+1} = x_{kj} + a_{kj}p_j$ minimizes $F(x_{xj} + \alpha p_j)$.

When this algorithm for computing $\bar{x}_k$ in (24e) is used, the method (24) of parallel plane is carried out by successive minimization along suitably chosen lines.

The method algorithm (24) combined with (25) can be used for finding the minimum of a nonquadratic function of class C''. In this case one restarts after the point x_{n+1} has been obtained. The point x_{n+1} as the new initial point x_1. In this manner one can obtain an algorithm based on the computation of functional values only. Methods of this nature have been used by Powell, Zangwill and Chazan and Miranker.

8. METHOD OF PARALLEL DISPLACEMENTS. If in the algorithm (24) combined with (25) we select the point $\hat{x}_k$ to be the initial point x_1 the computation can be carried out in parallel so as to yield the following algorithm. Select (n+1)-independent points $x_1, x_{11}, x_{21}, \ldots, x_{n1}$, that is, points such that the vectors $u_k = x_{k1} - x_1$ $(k = 1, \ldots, n)$ are linearly independent. Select $p_1 = x_{11} - x_1$ and compute the minimum points $x_2 = x_{12}, x_{22}, x_{32}, \ldots, x_{n2}$ of F on the n parallel lines

$$x = x_{j1} + \alpha p_1 \qquad (j = 1, \ldots, n).$$

Observe that the points $x_2, x_{22}, x_{32}, \ldots, x_{n2}$ are independent points which determine the (n-1)-plane π_{n-1} conjugate to p_1 passing through the minimum point x_0 of F. We now repeat our procedure in π_{n-1} and set $p_2 = x_{22} - x_2$ and compute the minimum points $x_3 = x_{23}, x_{33}, x_{43}, \ldots, x_{n3}$ of F on the n-1 parallel lines

$$x = x_{j2} + \alpha p_2 \qquad (j = 2, \ldots, n).$$

The (n-1) points $x_3, x_{33}, \ldots, x_{3n}$ are independent and determine the (n-2)-plane π_{n-2} conjugate to p_1 and p_2. Again the minimum point of F lies in π_{n-2}. Continuing in this manner we obtain the minimum point $x_{n+1} = x_0$ of F in n-steps. The iteration is illustrated schematically for the case n = 3.

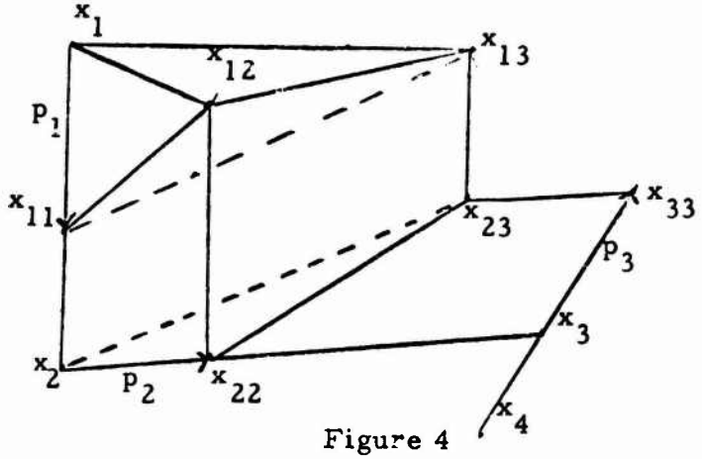


Figure 4

It is clear that this algorithm can be modified so as to be applicable to nonquadratic functions.

9. CONJUGATE GRAM-SCHMIDT PROCESSES. The parallel plane algorithm (24) described above is not in the most convenient computational form. In order to obtain a more convenient form observe that by virtue of (24f) with $\hat{x}_k = x_1$ the vectors $p_1, p_2, \ldots,$ are given by the algorithm

(26a) $$p_1 = u_1, \; p_k = u_{k1} p_1 + \cdots + b_{k,k-1} p_{k-1}.$$

where b_{kj} is chosen so that p_k is conjugate to p_j. We have accordingly

(26b) $$b_{kj} = -\frac{u_k^* A p_j}{d_j}, \quad d_j = p_j^* A p_j \quad (j = 1, \ldots, k-1).$$

The equations (26) define a conjugate Gram Schmidt Process for transforming a set of n linearly independent vectors $u_1, \ldots, u_n$ into a set of n mutually conjugate vectors $p_1, \ldots, p_n$. Using this procedure to generate the mutually

conjugate vectors appearing in the conjugate direction algorithm we obtain the following algorithm which we shall call the conjugate Gram Schmidt process (CGS-process).

(i) Starting with an initial point x_1 and an initial direction $p_1 = u_1$ compute

(27a) $$r_1 = -F'(x_1)\ ,\ s_1 = Ap_1, c_1 = p_1^* r_1,\ d_1 = p_1^* s_1\ ,\ a_1 = c_1/d_1$$

and obtain the point

(27b) $$x_2 = x_1 + a_1 p_1$$

(ii) For $k = 2, 3, \ldots$ choose u_k such that $u_1, \ldots, u_k$ are linearly independent. Compute

(27c) $$b_{kj} = -u_k^* s_j / d_j \qquad (j = 1, \ldots, k-1)$$

(27d) $$p_k = u_k + \sum_{j=1}^{k-1} b_{kj} p_j\ ,\ s_k = Ap_k$$

(27e) $$c_k = p_k^* r_1,\ d_k = p_k^* s_k\ ,\ a_k = c_k/d_k$$

and find the next point

(27f) $$x_{k+1} = x_k + a_k p_k .$$

The point x_{n+1} obtained in this manner is the minimum point of our quadratic function F.

Observe that the vector s_k appearing in this algorithm can be computed by the formula

(28) $$s_k = \frac{F'(x_1 + \sigma_k p_k) - F'(x_1)}{\sigma_k}\ ,\ \sigma_k = \frac{\sigma}{|p_1|}$$

where σ is constant. When this formula for s_k is used the CGS-algorithm is applicable to nonquadratic functions. As in the cg-algorithm we restart after n steps. One can modify the algorithm still further so as to express c_k, d_k, a_k, b_{kj} in terms of functional values only. When this is done one obtains an effective algorithm for minimizing a function without computation of derivatives. This algorithm has been used effectively by Dennemeyer and Maskini.

10. MATRICES ASSOCIATED WITH MUTUALLY CONJUGATE VECTORS. Let A be the positive definite matrix associated with the quadratic function F. Let H be a second positive matrix. We begin with the following result.

Lemma. If p is a nonnull vector, $s = Ap$, $q = Hs$, $d = p^*s$, $\delta = q^*s$ and $\alpha = d/\delta$, then the vector

$$\hat{p} = p - \alpha q$$

is conjugate to p. Moreover if $\sigma \geq 0$ the matrix

$$\hat{H} = H - \frac{qq^*}{\delta} + \frac{pp^*}{d} + \sigma \frac{pp^*}{d} \tag{29}$$

is a positive definite matrix such that

$$\hat{H}s = \hat{H}Ap = p . \tag{30}$$

Since $p^*A\hat{p} = s^*(p - \alpha q) = d - \alpha\delta = 0$ it follows that $\hat{p}$ is conjugate to p and orthogonal to s. By the computations

$$Hs = Hs - \frac{qq^*s}{\delta} + \frac{pp^*s}{d} + \frac{\hat{p}\hat{p}^*s}{d} = q - q + p = p$$

it is seen that (30) holds. Observe that the matrix

$$M = H - \frac{qq^*}{\delta}$$

has the rank $n-1$ inasmuch as $Ms = q - q = 0$. Hence $v^*Mv > 0$ unless v is a multiple of s. From this result we see that

$$v^*Hv = v^*Mv + \frac{(v^*p)^2}{d} + \sigma \frac{(v^*\hat{p})^2}{d}$$

is positive when v is not a multiple of s. If $v = \beta s$ $(\beta \neq 0)$ it has the value $\beta^2 d > 0$. Hence $\hat{H}$ is positive definite and the lemma is established.

It is clear that if $\hat{p} = 0$, then $\hat{H}$ is positive definite for all values of σ. If $\hat{p} \neq 0$ there is a least number σ_o such that $\hat{H}$ is positive definite whenever $\sigma > \sigma_o$. However we shall restrict ourselves to the case in which $\sigma \geq 0$. Observe that the matrix $\hat{H}$ is unaltered if we replace p by ρp, where ρ is a nonnull scale factor.

Of special interest is the case $\sigma = 0$ and $\sigma = \delta/d$. In these cases $\hat{H}$ is given by the formulas

$$\hat{H} = H - \frac{qq^*}{\delta} + \frac{pp^*}{d} \qquad \text{(case } \sigma = 0\text{)} \tag{31}$$

$$H = H - \frac{pq^* + qp^*}{d} + \frac{pp^*}{d} \qquad \text{(case } \sigma = \delta/d\text{)} \tag{32}$$

Given a set of mutually conjugate vectors $p_1, \ldots, p_n$, a set of nonnegative numbers $\sigma_1, \ldots, \sigma_n$ and a positive definite matrix H the algorithm

(33a) $$H_1 = H\ ,\ H_{k+1} = H_k - \frac{q_k q_k^*}{\delta_k} + \frac{p_k p_k^*}{d_k} + \sigma_k \frac{\hat{p}_{k+1}\hat{p}_{k+1}^*}{d_k}$$

with

(33b) $$s_k = Ap_k,\ q_k = H_k s_k\ ,\ d_k = p_k^* s_k\ ,\ \delta_k^* = q_k^* s_k\ ,\ \alpha_k = d_k/\delta_k$$

(33c) $$\hat{p}_{k+1} = p_k - \alpha_k q_k$$

defines a sequence of positive definite matrices $H_1 = H, H_2, \ldots, H_{n+1}$ such that

(34) $$H_{k+1} s_j = H_{k+1} A p_j = p_j \qquad (j = 1, \cdots, k)$$

and hence such that $H_{n+1} = A^{-1}$. The fact that H_{k+1} is positive definite follows from the last lemma. This lemma also tells us that (34) holds when $j = k$. Hence (34) holds when $k = 1$. If it holds when $k < m$ we have for $j < m$

$$q_m^* s_j = s_m^* H_m s_j = s_m^* p_j = 0,\ \hat{p}_m^* s_j = p_m^* s_j - \alpha_m q_m^* s_j = 0$$

$$H_{m+1} s_j = H_m s_j - \frac{q_m q_m^* s_j}{\delta_n} + \frac{p_m p_m^* s_j}{d_j} + \sigma_m \frac{\hat{p}_m \hat{p}_m^* s_j}{d_j} = p_j\ ,$$

Hence (34) holds, as stated. As a consequence of (34) we have

(35) $$s_j^* q_{k+1} = 0\ ,\ s_j^* \hat{p}_{k+1} = 0\ (j \leq k)\ ,$$

(36) $$s_j^* H_{k+1} v = 0 \text{ whenever } p_j^* v = 0.$$

In addition if v_k is a vector such that $p_k = H_k v_k$ then v_k is orthogonal to $p_1, \ldots, p_{k-1}$, as one readily verifies.

An interesting special case of the algorithm (33) is the case in which $\sigma_k = 0$ and p_{k+1} coincides with $\hat{p}_{k+1}$. In this event we have

(37a) $$p_{k+1} = p_k - \alpha_k q_k\ ,\ \alpha_k = d_k/\delta_k,\ s_{k+1} = Ap_{k+1}$$

(37b) $$q_{k+1} = H_{k+1} s_{k+1} = Hs_{k+1} + \beta_k q_k\ ,\ \beta_k = -\frac{s_k^* H s_{k+1}}{\delta_k}$$

with $q_1 = Hs_1 = HAp_1$ initially. This is one form of the conjugate gradient algorithm for generating mutually conjugate vectors. If $r_1 = -F'(x_1)$, $p_1 = Hr_1$ and $r_2, r_3, \cdots$ are generated by the formulas

(38) $$r_{k+1} = r_k - a_k\,,\quad a_k = c_k/d_k\,,\quad c_k = p_k^* r_k$$

then p_{k+1} is given by the alternative formula

(39) $$p_{k+1} = H_{k+1}\, r_{k+1}.$$

This can be seen by induction with the help of equations (33), (37) and (38). From these results we obtain the following form of the conjugate gradient algorithm.

Given an initial point x_1 and a positive definite matrix H set $r_1 = -F'(x_1)$, $p_1 = Hr_1$, $H_1 = H$ proceed in the k-th step as follows

(40a) Choose $x_{k+1} = x_k + a_k p_k$ so as to minimize $F(x_k + a p_k)$ and set

(40b) $$s_k = \frac{F'(x_{k+1}) - F'(x_k)}{a_k}$$

(40c) $$q_k = H_k s_k\,,\quad \delta_k = q_k^* s_k\,,\quad d_k = p_k^* s_k$$

(40d) $$H_{k+1} = H_k - \frac{q_k q_k^*}{\delta_k} + \frac{p_k p_k^*}{d_k}\,,\quad p_{k+1} = -H_{k+1} F'(x_{k+1})\,.$$

This form of the cg-algorithm is applicable at once to nonquadratic case by introducing method for minimizing $F(x_k + a p_k)$. In the quadratic case this minimum can be obtained by a formula. This algorithm is known as the Davidon-Fletcher-Poweli algorithm. It was suggested originally by Davidon. If we use the formula (33a) for H_{k+1} with $\sigma_k > 0$ we obtain the same directions $p_1, p_2, \ldots$. However their lengths have been altered.

An alternate form is obtained by replacing (40a) and (40b) by the computations

$$s_k = \frac{F'(x_k + \sigma_k p_k) - F'(x_k)}{\sigma_k}\,,\quad \sigma_k = \frac{\sigma}{|p_k|}$$

$$c_k = -p_k^* F'(x_1)\,,\quad d_k = p_k^* s_k\,,\quad a_k = c_k/d_k$$

$$x_{k+1} = x_k + a_k p_k.$$

Here σ is a small positive constant. In this modification the point $x_{k+1} = x_k + a_k p_k$ minimizes $F(x_k + a p_k)$ in the quadratic case and is an estimate of the minimum point in the nonquadratic case. This

procedure avoids search routines . Various rules can be given for the choice of σ.

In a similar manner the conjugate Gram Schmidt routine can be put in matrix form so as to obtain alternative algorithms for minimizing quadratic and nonquadratic functions. All these algorithms can be viewed as modifications of Newton's method and the secant method, and have similar convergence properties.

REFERENCES

1. Hestenes, M. R. and Stiefel, E., Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards, Vol. 49 (1952), pp. 409-436.

2. Hestenes, M.R., The conjugate gradient method for solving linear systems, Proceedings of the Sixth Symposium in applied mathematics, edited by J. H. Curtiss, American M. .ematical Society 1956.

3. Fletcher, R. and Powell, M.J.D., A rapidly convergent descent method for minimization, Computer Journal, Vol. 6, 1963, pp.163-165.

4. Powell, M.J.D., An efficient method for finding the minimum of a function of several variables without calculating derivatives, Computer Journal, Vol. 7 (1964), pp. 155-162.

5. Zangwill, W.I., Minimizing a function without calculating derivatives, Computer Journal, Vol. 10 (1967), pp. 293-296.

6. Chazan, D. and Miranker, W.L., A nongradient and parallel algorithm for unconstrained minimization, SIAM J. Control, Vol. 8 (1970), pp. 207-217.

7. Hestenes, M.R., Multiplier and gradient methods, Journal of Optimization theory and applications, Vol. 4(1969), pp. 303-320.

8. Dennemeyer, R. and Mookini, E., CGS algorithms for unconstrained minimization of functions, to appear in Journal of Optimization theory and applications.

COMPUTERGRAPHICS LANGUAGE FOR YOUR DESIGN EQUATIONS

(CLYDE)

R. I. ISAKOWER and R. E. BARNAS

Scientific and Engineering Application Division
Management information Systems Directorate
Picatinny Arsenal, Dover, New Jersey

ABSTRACT

CLYDE is a computergraphics language for your design equations. It is the aftermath of the PDQ series, providing an interactive graphics solution to an important group of second and fourth order partial differential equations. These equations appear in almost every branch of applied mathematics: governing the solutions to design problems in heat transfer, stress analysis, and potential fields (electric, magnetic, electrostatic, gravitation, velocity in irrotational flow, etc . . .). This document is intended as a press release - to pictorially reveal the diverse engineering applications available. CLYDE was written for a CDC 6500/1700/274 facility operating under SCOPE 3.3, IGS V.2 employing 32 overlays and 50K bytes of storage.

THE PROBLEM

a. Background and Capabilities:

Most munitions design is governed by the classical ideas and equations of continuum mechanics. Through the descriptive differential equations of elasticity, classical mechanics, electromagnetic theory, fluid mechanics, etc., the working state of munition items may be accurately studied. Physical phenomena in continous systems - elastic bodies, fluids - are usually described by partial differential equations with their associated boundary or initial conditions. Closed form solutions to these PDE's are rarely available in the design room with its configurations that perversely do not conform to classical text book illustrations. Therefore, in the harsh work of reality, recourse to numerical solutions is an absolute necessity. This document describes the latest interactive graphics implementation of a numerical solution (finite differences) to an important class of boundary value problems involving the second order equations

$$\nabla^2 f = A\frac{\partial^2 f}{\partial x^2} + B\frac{\partial^2 f}{\partial y^2} = D(x, y)$$

$$\nabla^2 f = A\frac{\partial^2 f}{\partial z^2} + B\frac{\partial^2 f}{\partial r^2} + \frac{C}{r}\frac{\partial f}{\partial r} = D(r, z)$$

and the fourth order equation.

$$\nabla^4 f = \frac{\partial^4 f}{\partial x^4} + 2\frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4} = g(x, y)$$

Finite difference approximations are made of the derivatives, in either cartesian or cylindrical coordinates, employing a generalized irregular star for the "computation stencil". Program options include variable grid size, curved boundaries, variable boundary conditions, and domain loading (stress analysis), generalized equation coefficients, "mirror" boundary edges for repeating sections punched deck and print-out of results, and hard copy (plotter) of problem domain with contour maps of selected values of the solved variable. In addition, the user may specify a finer mesh size in critical regions, modify the scale of the problem picture, change the boundary of the problem area (or redraw it completely), and change boundary conditions and problem loading. It is possible to pass a plane (shown as a straight line) through the picture of the problem. The variation or plot of the solved variable along that plane is displayed on the screen.

Use of the program is illustrated with problems in steady state temperature distribution and stress analysis of laterally loaded flat plates. Addition examples are shown in the Appendix - of particular interest is the membrane or soap film analogy of the torsion of bars and shafts.

b. General Method of Solutions:

The physical description of the problem to be solved is (generally) inputted to the computer program on punched cards. The picture of the problem then appears on the screen and the interactive graphics solution is airborne.

The computer program overlays the domain of the problem with a rectangular net of vertical and horizontal grid lines. To conserve the graphics screen refresh memory these grid lines are not displayed on the CRT, only their intersections. (Again to conserve refresh memory, only a "repeating section" is displayed, thus taking advantage of any symmetry

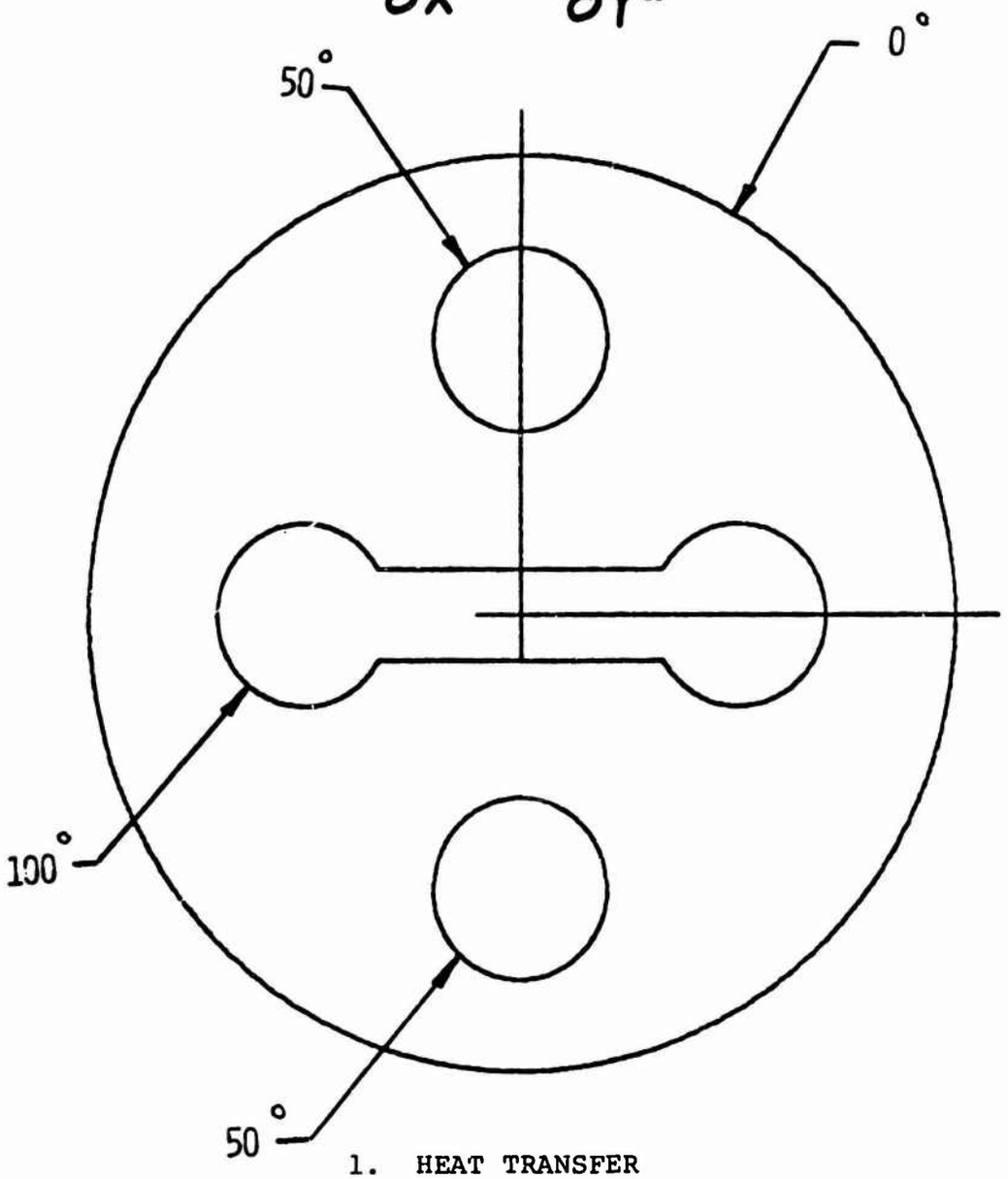
of the problem). The intersections of the grid lines with the boundaries of the problem are called boundary nodes and are shown as little circles. The intersections of the vertical and horizontal grid lines are called domain nodes and are shown as little crosses. It is at these little crosses, within the problem area, that the finite difference approximations (algebraic expressions) are applied. But first, the domain nodes (crosses) that are not within the problem area must be eliminated. A graphics command on the CRT is used to automatically eliminate most of the crosses outside the area. The light pen is then used to selectively eliminate those crosses inside holes in the problem and within overhanging portions of the problem area.

When the designer is satisfied that only the relevant domain nodes (crosses) are left, he then instructs the progam to solve the set of algebraic equations (one per node) that was used to approximate the original partial differential equation. The range of values of the parameter (just solved for) is flashed on the screen and the designer can display contour maps of desired values. A Calcomp plot of the full area is optionally available.

If desired, the graphics designer may redesign the problem at the screen (problem contour, boundary conditions, equation coefficients, etc), and resolve the "new" design problem.

Step-by-step illustrations of the above procedure to solve relevant design problems follow.

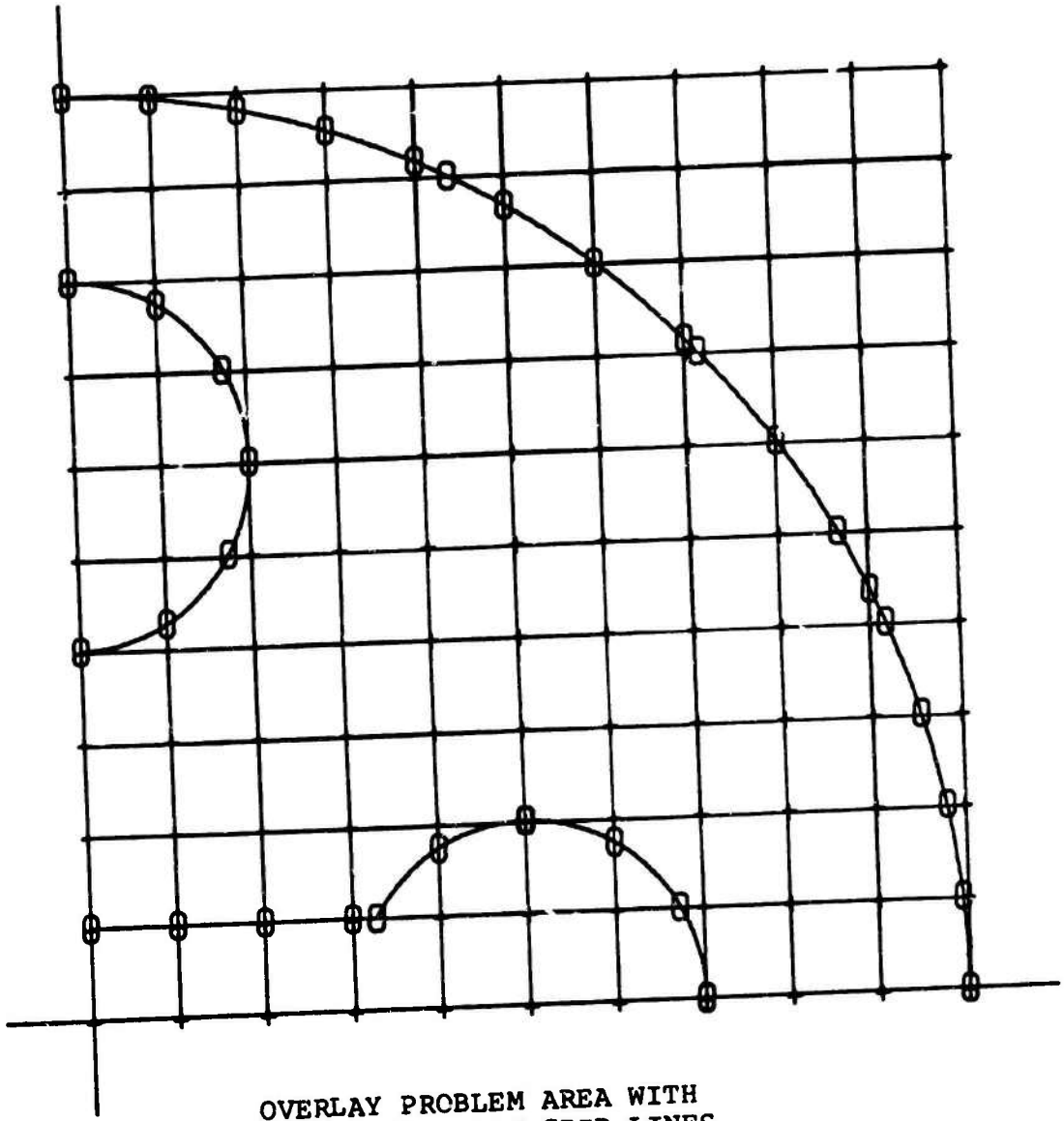
$$\frac{\partial^2 T}{\partial X^2} + \frac{\partial^2 T}{\partial Y^2} = 0$$



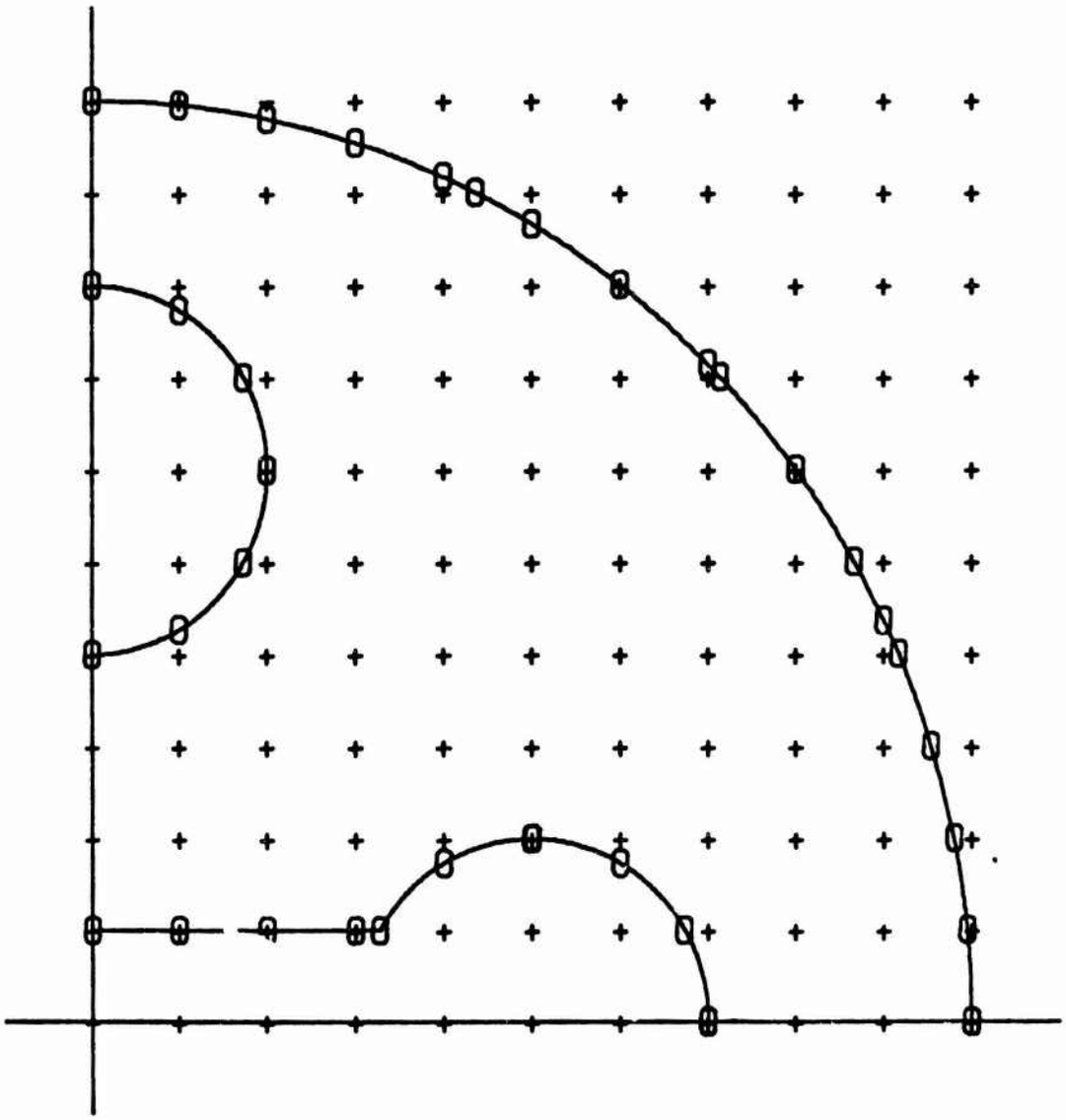
1. HEAT TRANSFER

The steady state temperature distribution in a circular plate with irregular perforations is studied on the following pages. The outer circumference of the plate is maintained at 0 degrees, the contours of the two inner circular perforations at 50 degrees, while the contour of the double-hole-slot perforation is constant at 100 degrees. The problem is to determine the temperature distribution throughout the plate.

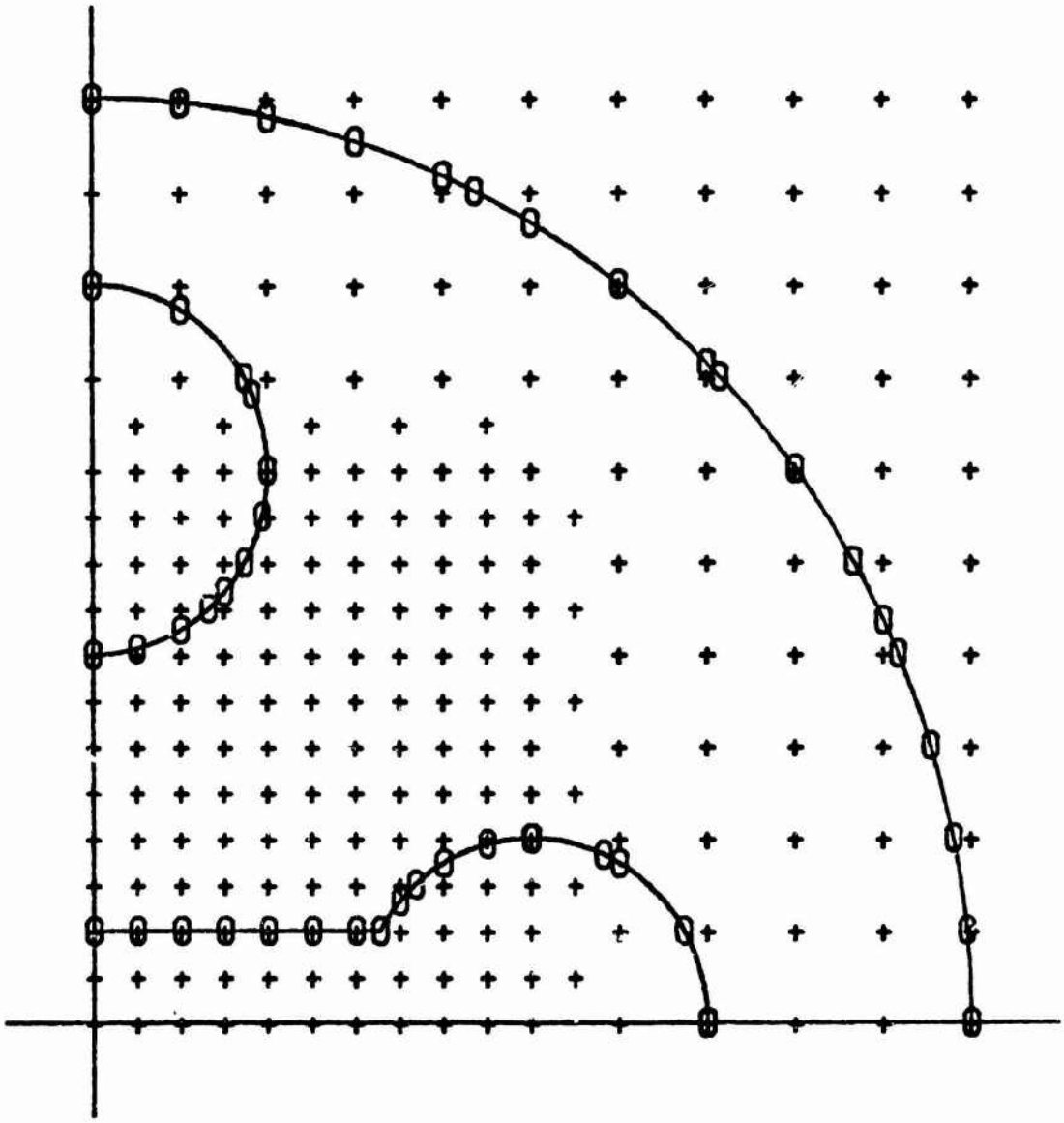
PLATE IS SYMMETRICAL. ONLY
ONE QUADRANT NEED BE EXAMINED.



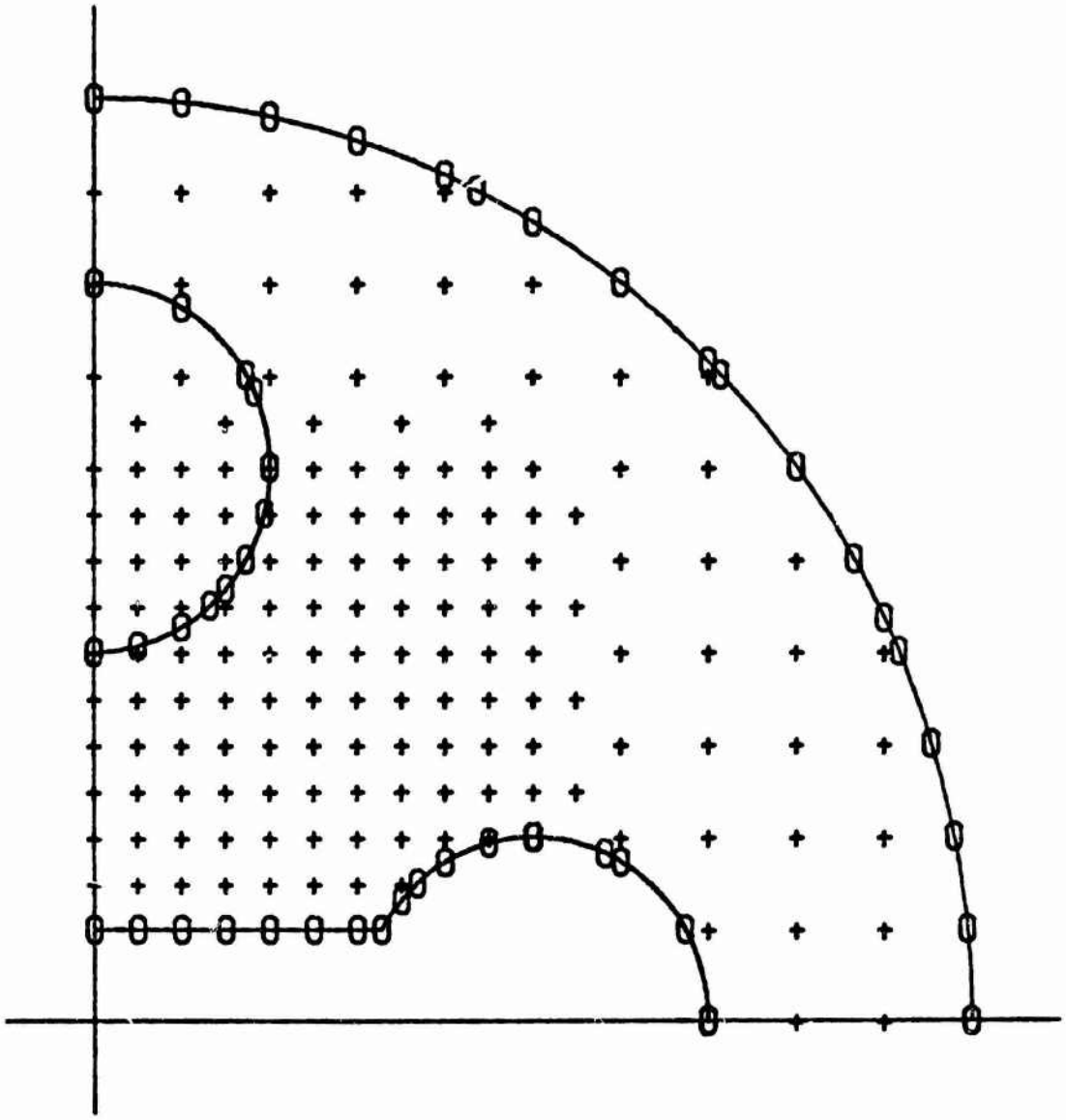
OVERLAY PROBLEM AREA WITH
RECTANGULAR NET OF GRID LINES.



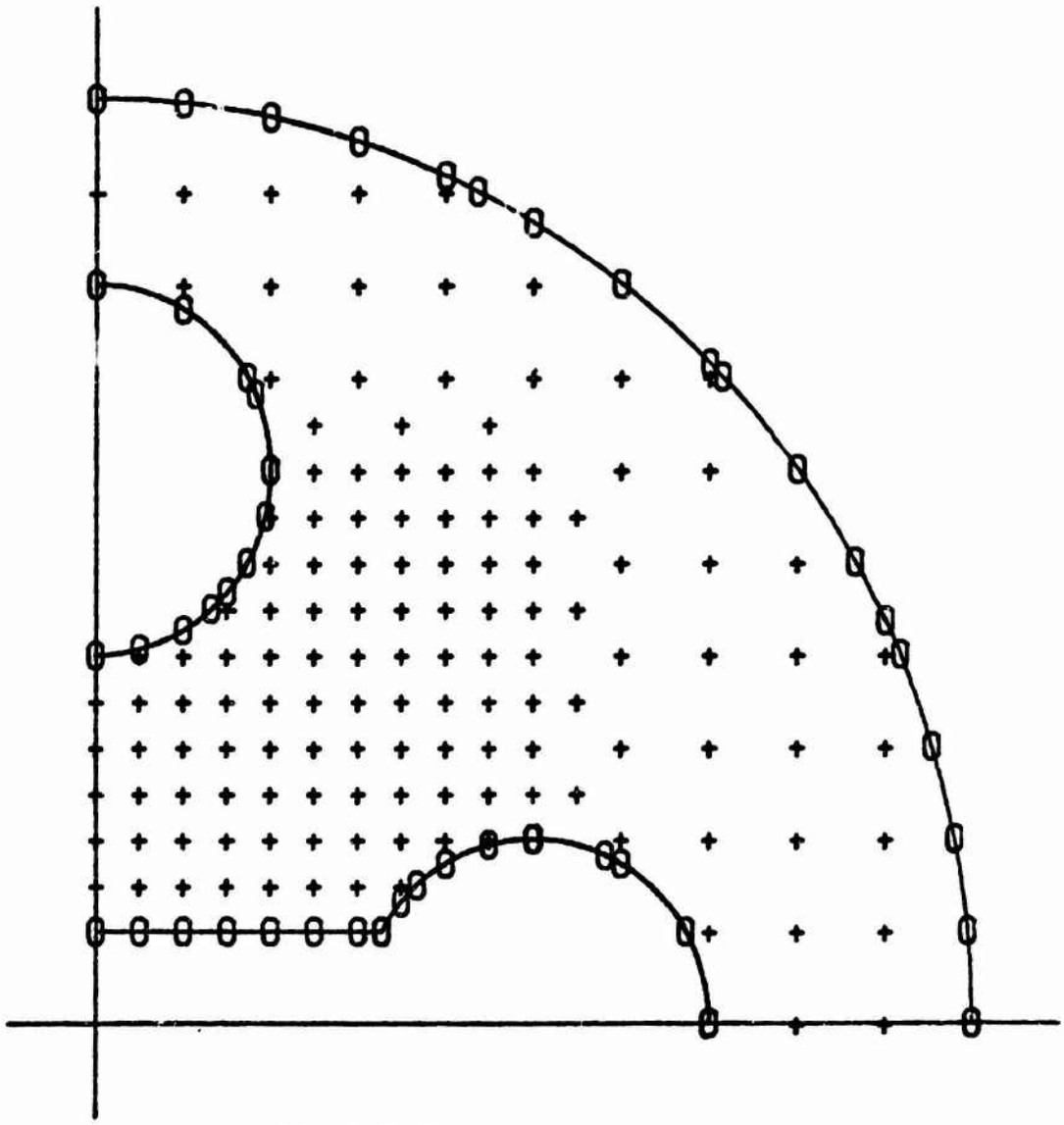
ONLY INTERSECTIONS OF
VERTICAL AND HORIZONTAL GRID
LINES ARE SHOWN (AS CROSSES).



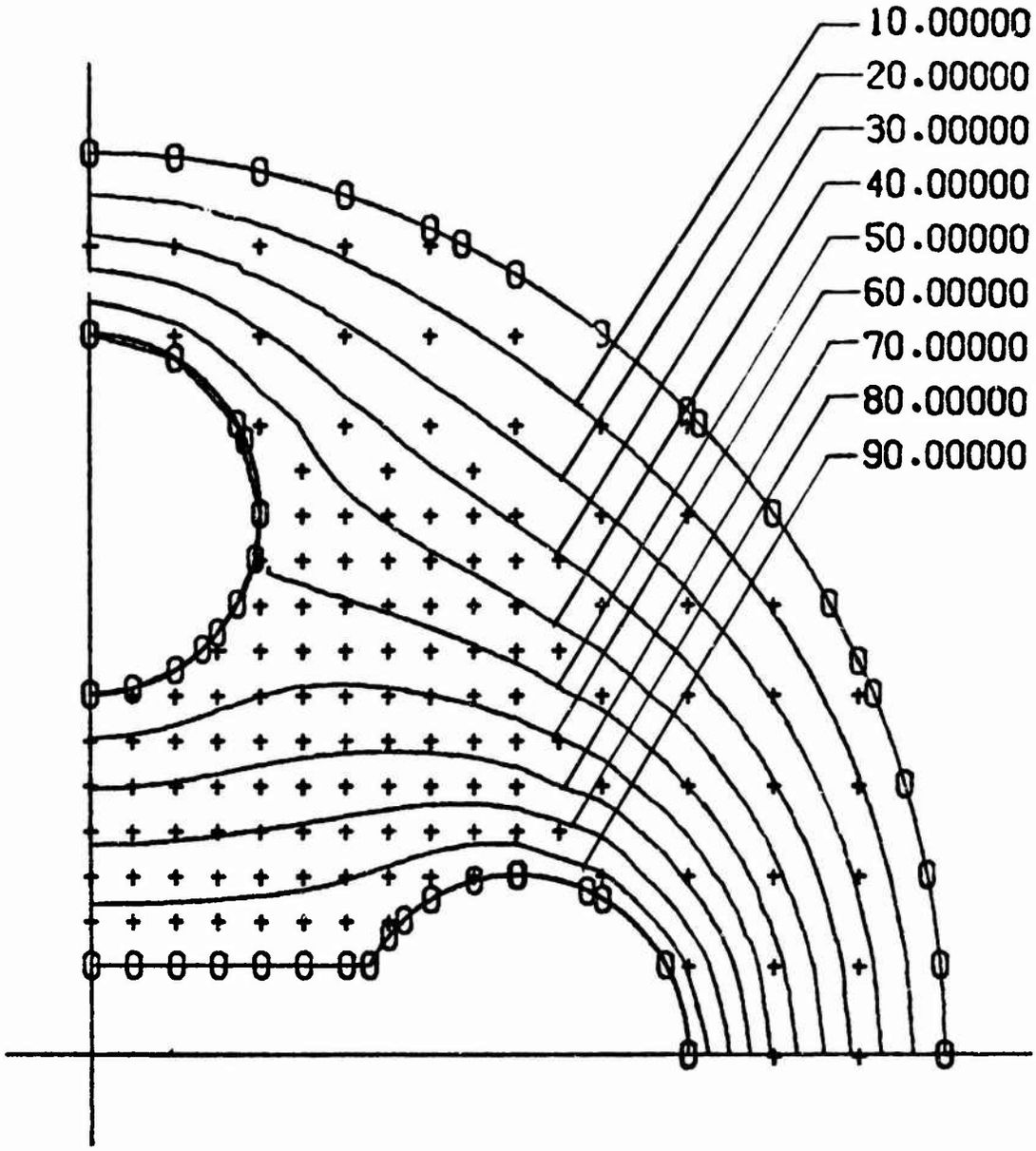
GAIN RESOLUTION WITH FINER
MESH OVER CRITICAL AREA.



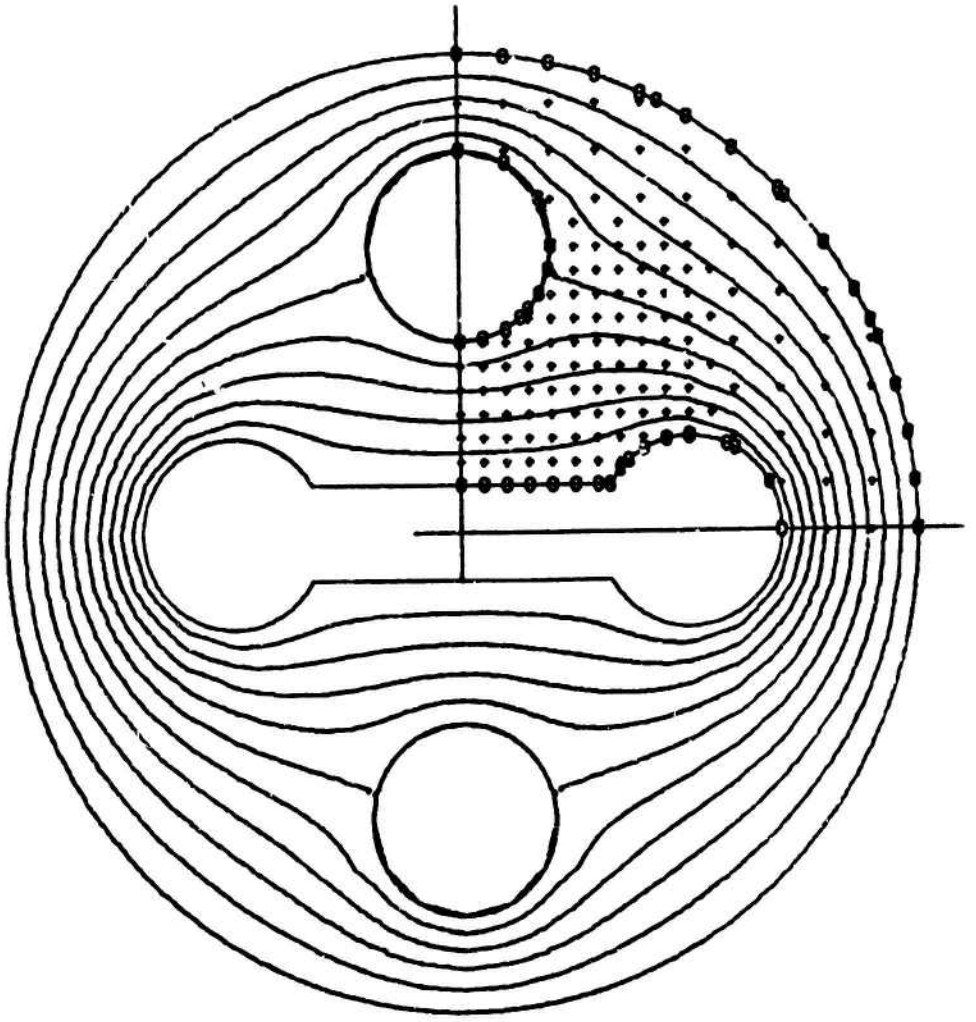
AUTOMATIC ELIMATION OF MOST
INNER DOMAIN NODES (CROSSES)
OUTSIDE OF PROBLEM AREA.



LIGHT PEN ELIMINATION OF
CROSSES IN HOLES & CREVICES.

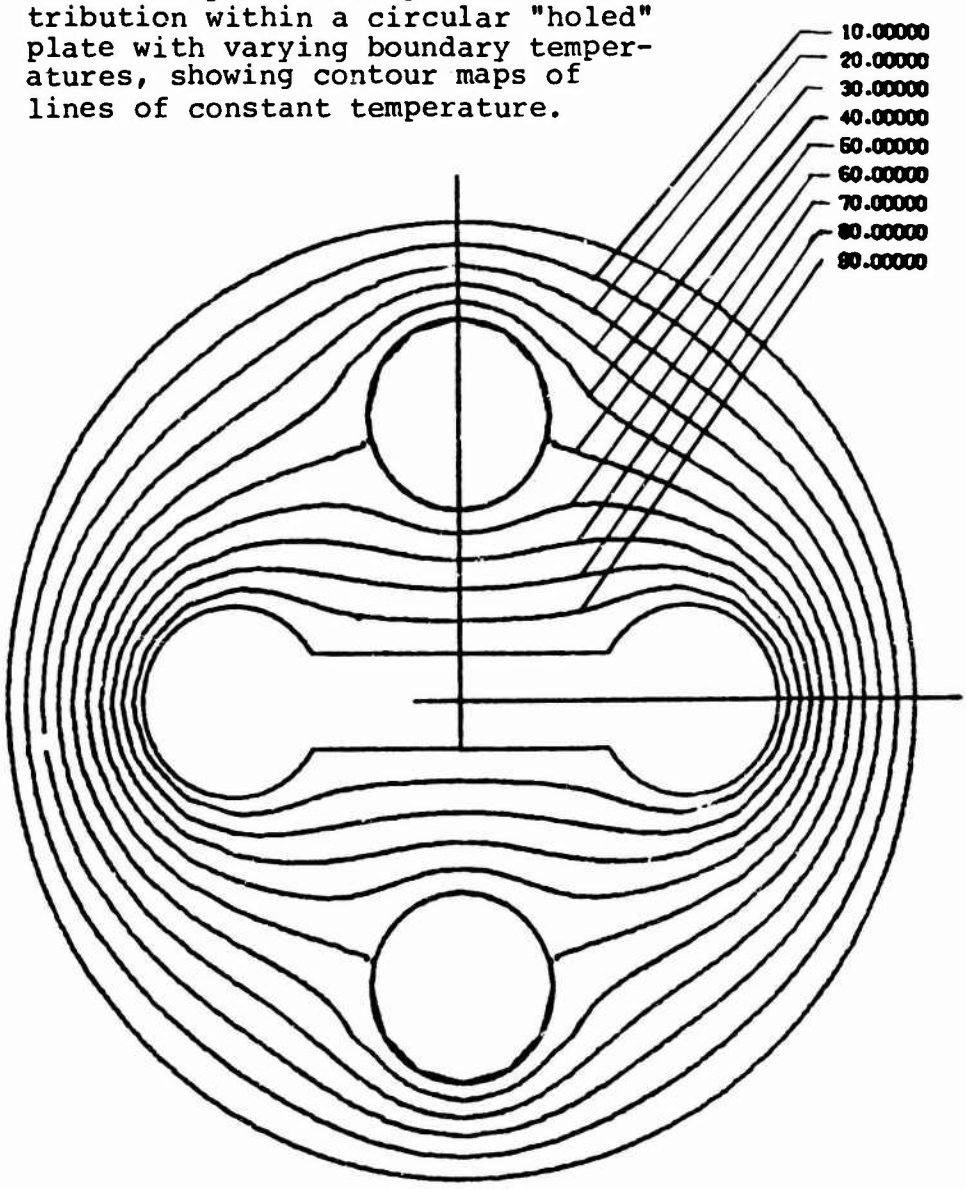


PROBLEM SOLVED, FOLLOWED BY
CONTOUR MAPPING DISPLAY OF
SELECTED VALUES.



CONTOUR MAP OF ENTIRE PLATE
WITH NODES DISPLAYED.

Steady State Temperature distribution within a circular "holed" plate with varying boundary temperatures, showing contour maps of lines of constant temperature.



CONTOUR MAP OF ENTIRE PLATE
WITH NODES SUPPRESSED.

2. Plate Stress Analysis

The deflection (w) of a thin plate loaded normal to its plane is described by the fourth order partial differential equation:

$$\frac{\partial^4 W}{\partial x^4} + 2\frac{\partial^4 W}{\partial x^2 \partial Y^2} + \frac{\partial^4 W}{\partial Y^4} = \frac{q}{D}$$

Unfortunately, because of the many configurations possible, a generalized finite difference operator for an irregular boundary problem perversely resists formulation. But despair ye not - no copout is forthcoming.....

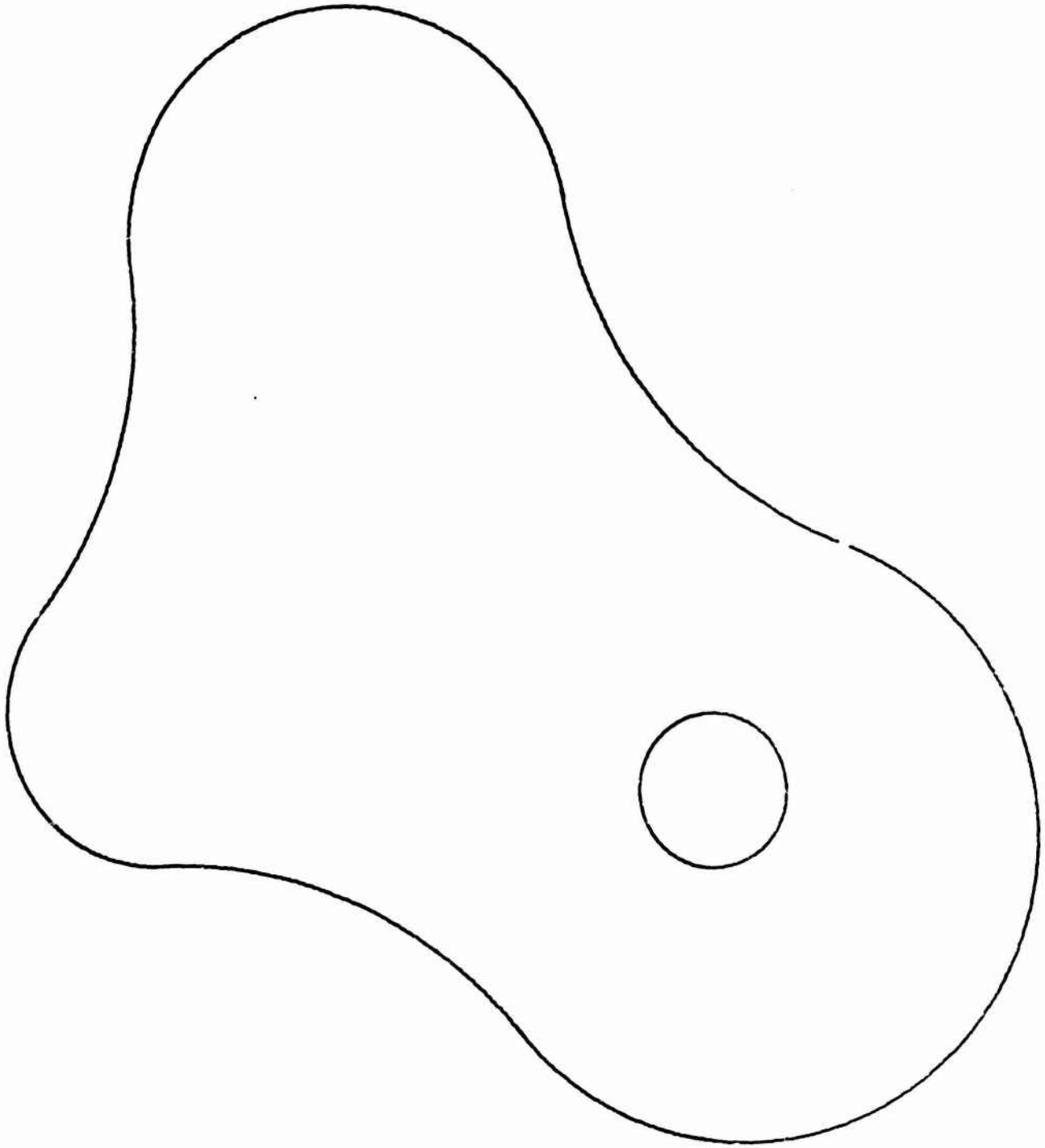
It is possible to replace the fourth order equation with two equations of the second order which represent the deflections of a membrane:

$$\frac{\partial^2 M}{\partial x^2} + \frac{\partial^2 M}{\partial Y^2} = -q$$

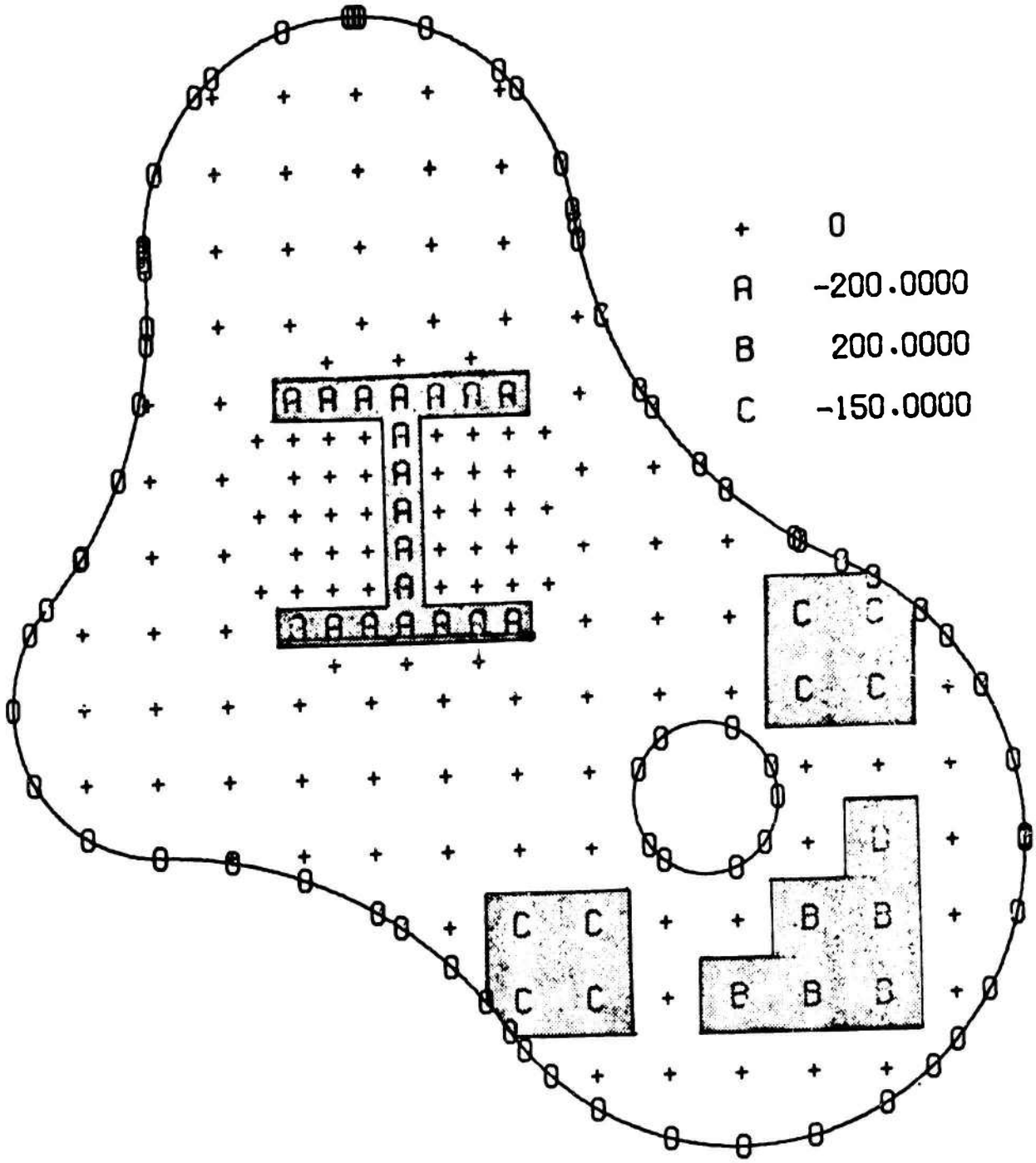
$$\frac{\partial^2 W}{\partial x^2} + \frac{\partial^2 W}{\partial Y^2} = -\frac{M}{D}$$

The two equations are solved sequentially (not simultaneously).

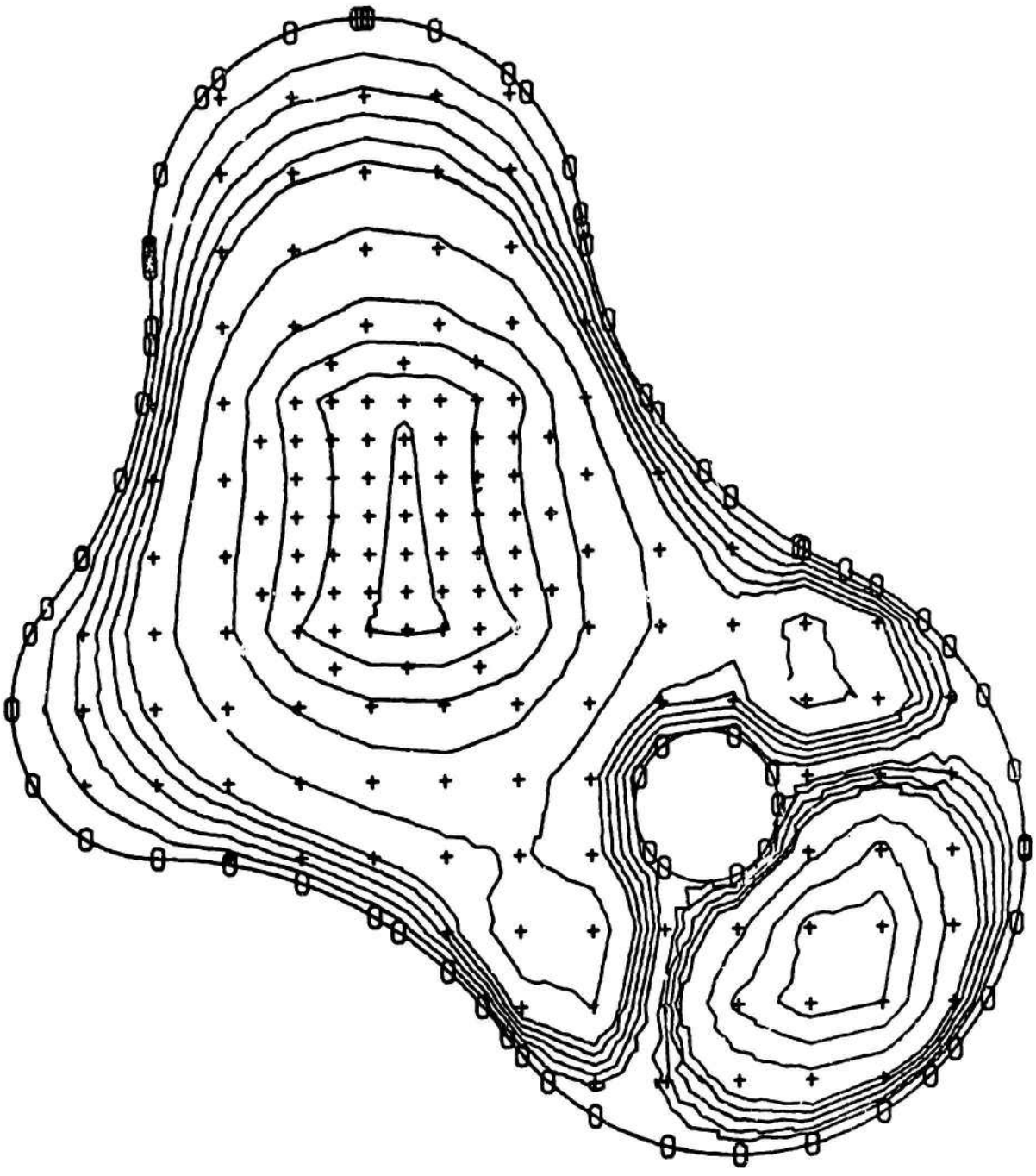
An arbitrarily contoured plate, simply supported at the edges, is loaded (at the Graphics screen) with three different valued loads. The hard copy output of the solution procedure follow.



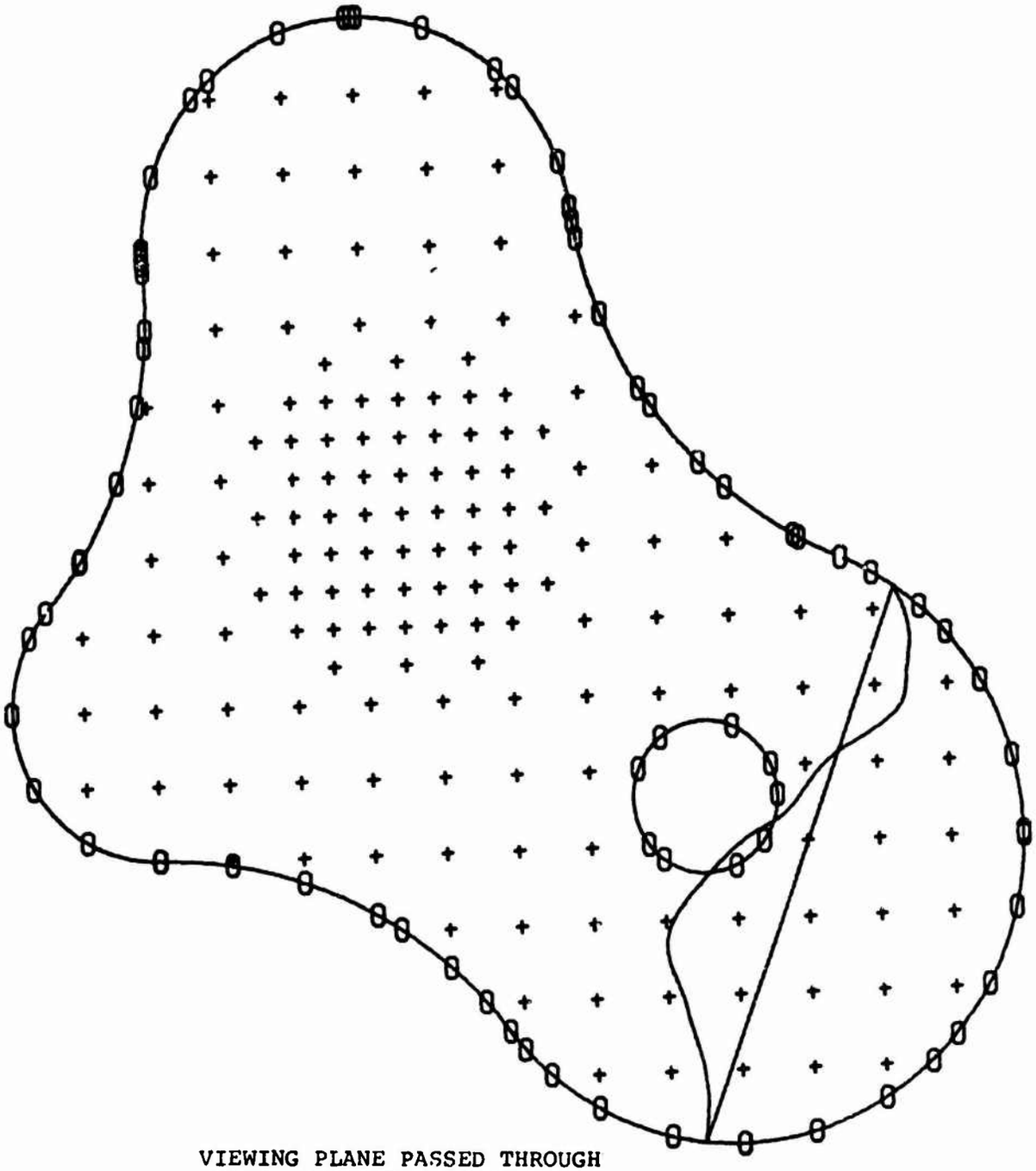
IRREGULAR FLAT PLATE



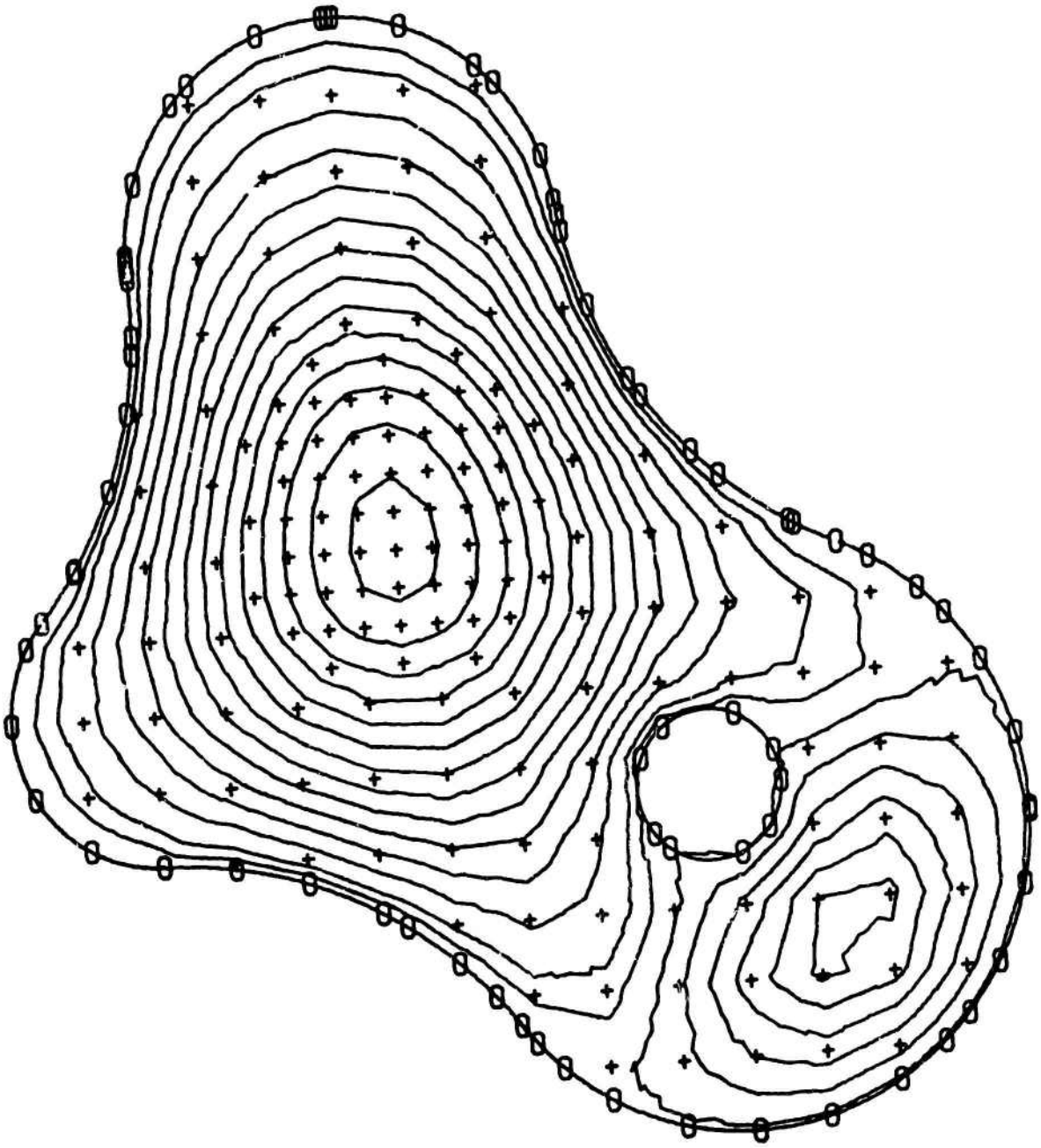
LOADING IS COMPLETE.
UP TO 26 DIFFERENT LOAD
VALUES ARE POSSIBLE.



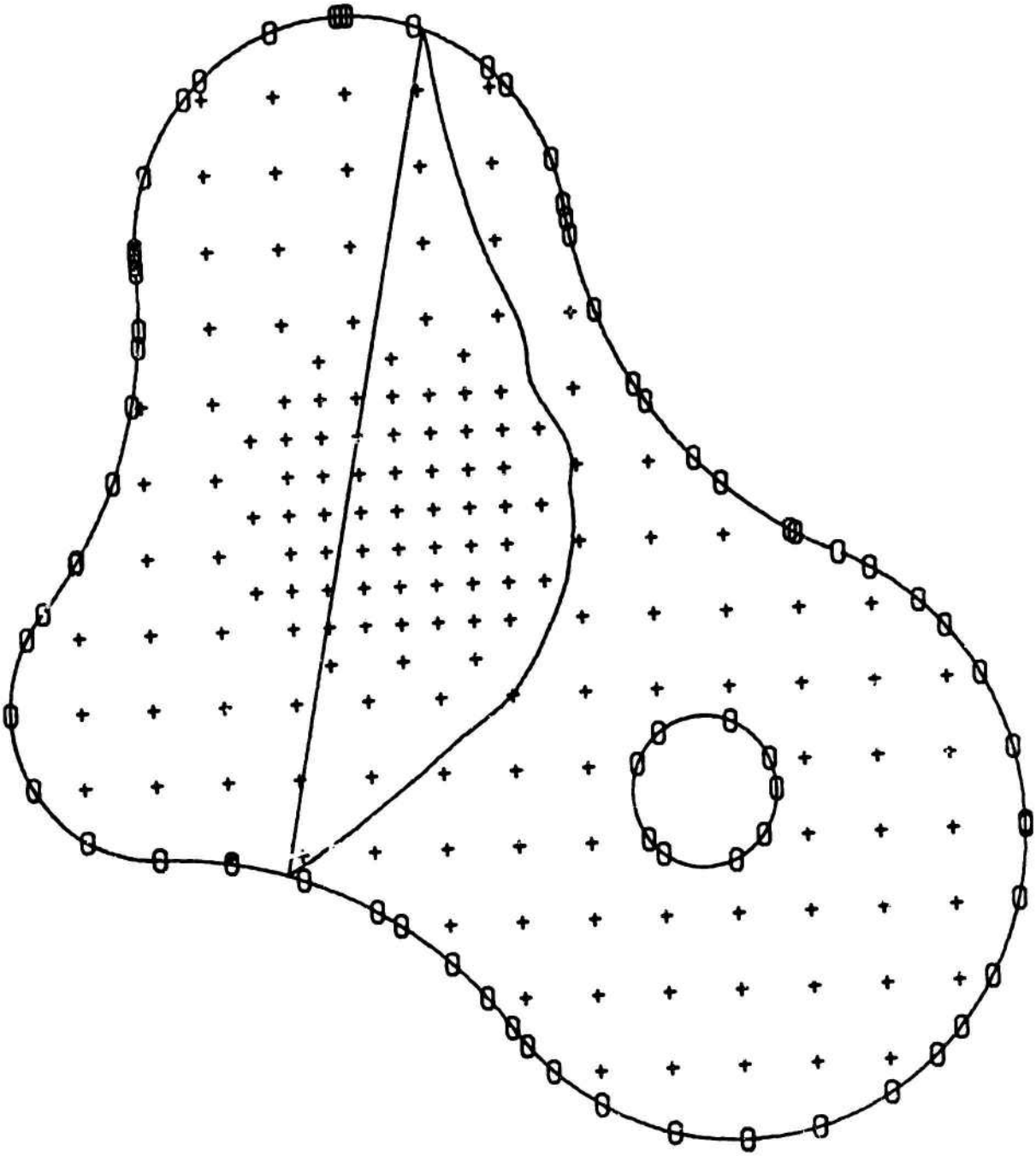
CONTOUR MAP OF SELECTED
RESULTANT PLATE MOMENTS.



VIEWING PLANE PASSED THROUGH PLATE. MOMENT DISTRIBUTION ALONG THAT PLANE IS "SCALED UP" FOR VIEWING PURPOSES.

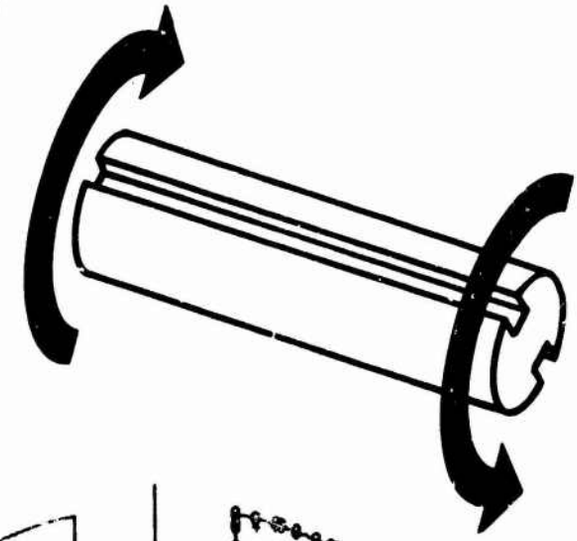


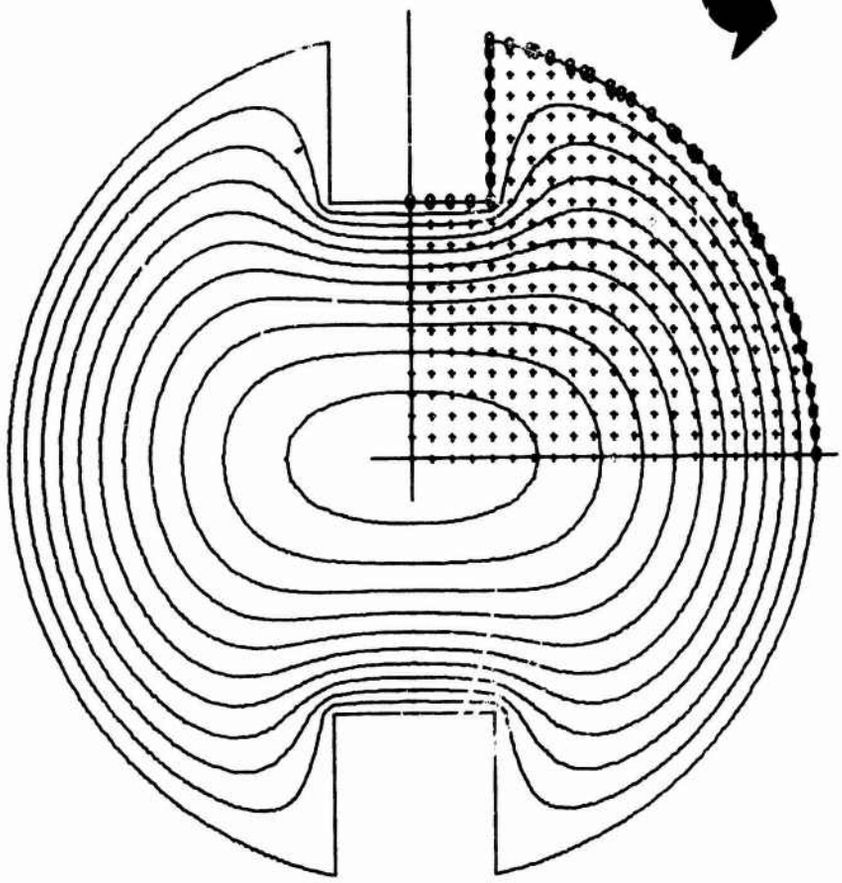
CONTOUR MAP OF RESULTANT
PLATE DEFLECTION.



AGAIN A VIEWING PLANE IS PASSED THROUGH THE PLATE... AND NOW THE DEFLECTION IS DISPLAYED "SCALED UP".

THE DEPENDENT VARIABLE IN THIS SHAFT TORSION EXAMPLE IS THE AIRY STRESS FUNCTION (φ). THE RATE OF CHANGE OF THIS STRESS FUNCTION IS DIRECTLY PROPORTIONAL TO THE TWIST-INDUCED SHEAR STRESS. CONTOUR LINES OF SOLVED CONSTANT STRESS FUNCTION VALUES WERE PLOTTED AT EQUALLY SPACED VALUES OF THE AIRY STRESS FUNCTION. THEREFORE, THE SPACINGS OF THE LINES ARE INVERSELY PORPORTIONAL TO THE SHEAR STRESSES. IN OTHER WORDS, THE CLOSER THE LINES - THE HIGHER THE SHEAR STRESS.

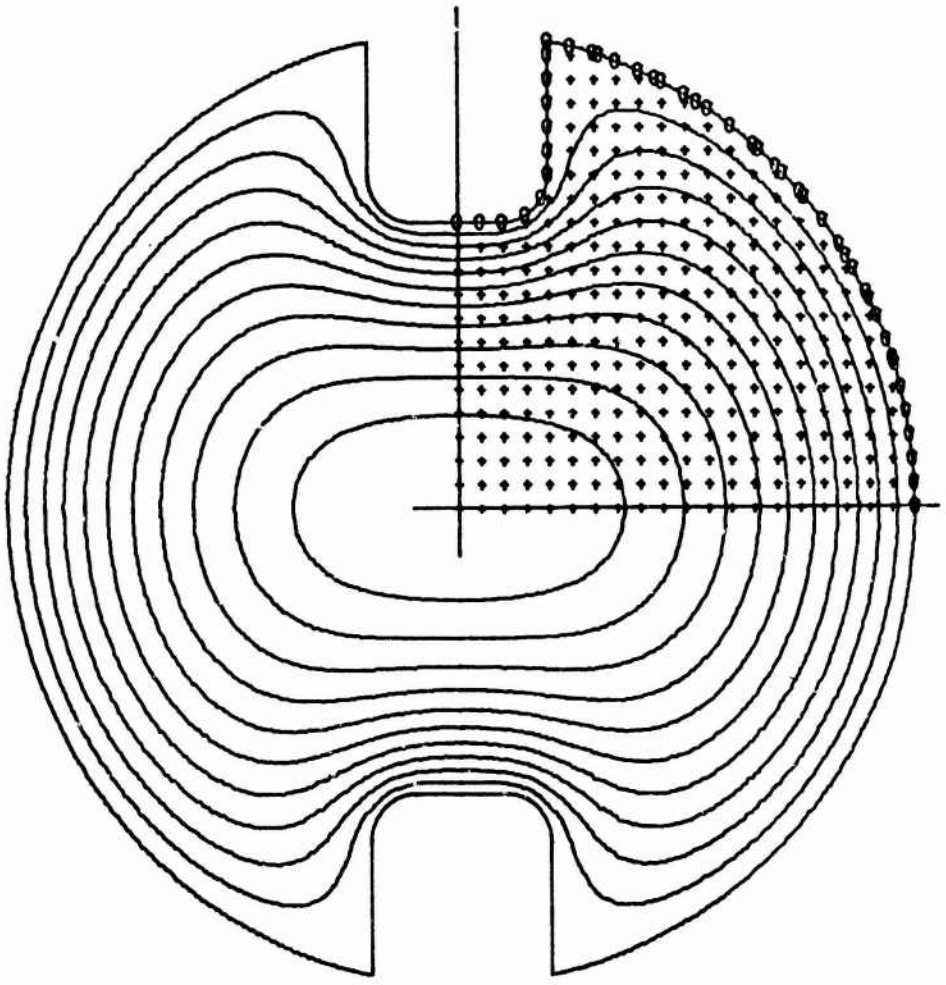




TORSION OF PRISMATIC BAR

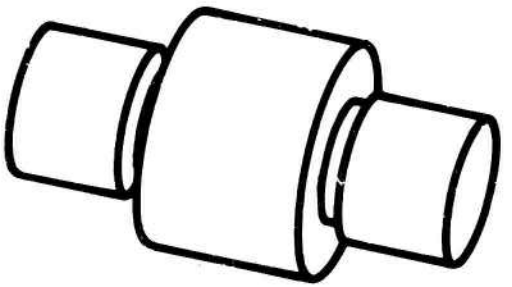
$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = -2$$

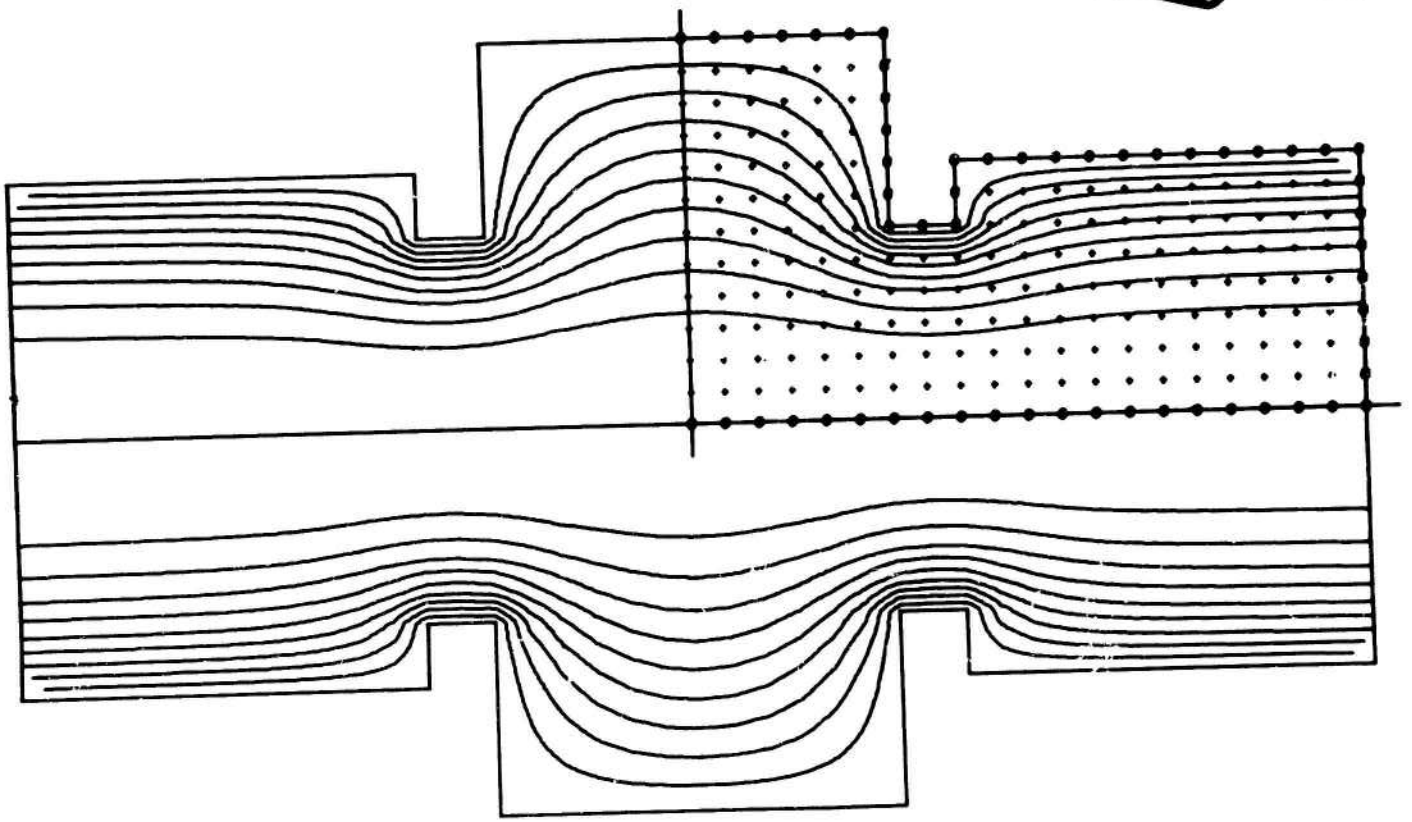
IN THE FIRST MAP IT MAY BE SEEN THAT THE INTERIOR RE-ENTRANT CORNERS OF THE KEYWAYS ARE STRESS RAISERS. THE SECOND MAP SHOWS A MODIFICATION OF THE SHAFT CONTOUR AND THE RESULTING (IMPROVED) STRESS FLOW. THE CONTOUR MODIFICATION WAS EFFECTUATED AT THE CONSOLE BY INSERTING FILLETS INTO THE KEYWAYS' CORNERS.



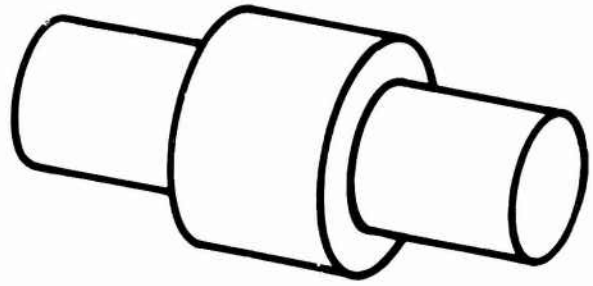
REDESIGNED SHAFT OUTPUT

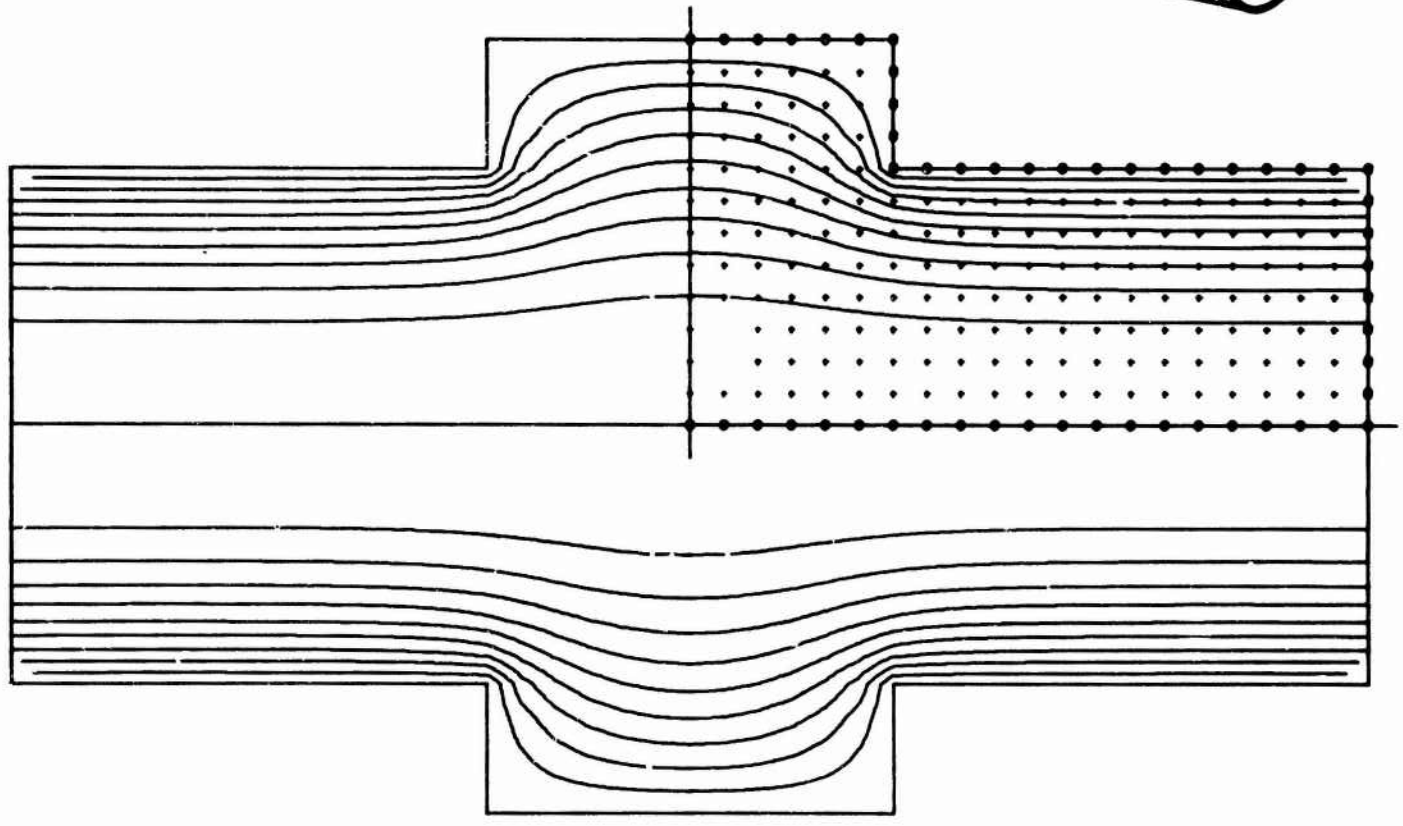
Contour map of stress functions in grooved and stepped shaft in torsion. (Cylindrical Coordinates).



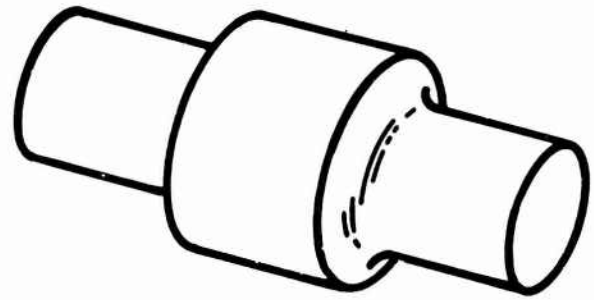


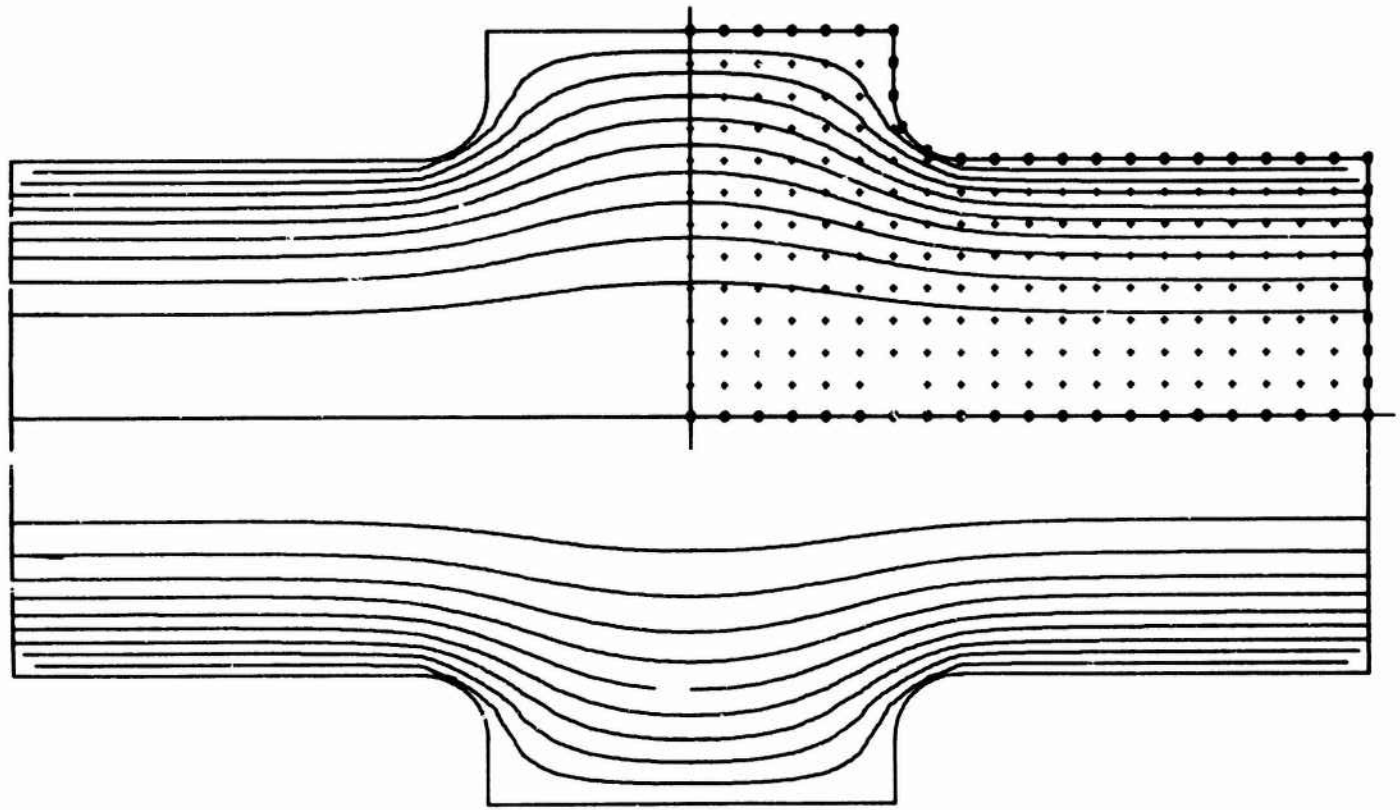
Torsion in redesigned stepped shaft (grooves removed).



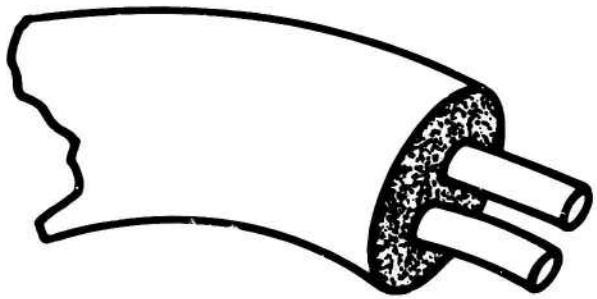


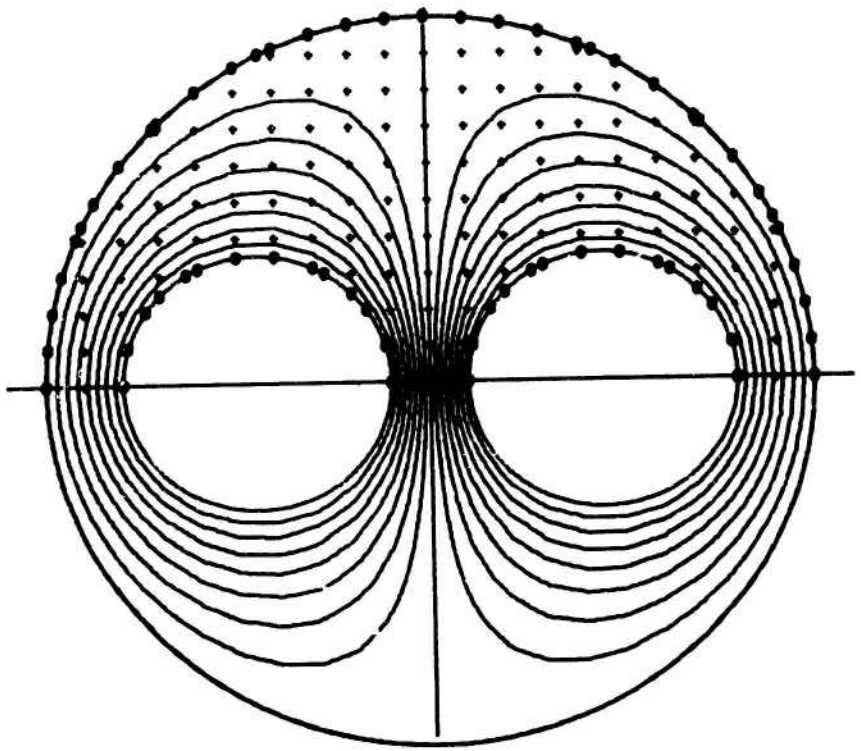
Final shaft design with a fillet between the two diameters shows a reduction in shear stress at the shoulder.





Experimental twin lead cable with contour map of lines of constant electrical potential.





The intersection of the grid lines with the boundaries of the domain are called boundary nodes. The intersections of the grid lines with each other within the problem domain are called inner domain nodes. It is at these inner domain nodes that the finite difference approximations are applied. The approximation of the partial differential equation with the proper finite difference operators replaces the PDE with a set of subsidiary linear algebraic equations - one at each inner domain node. For the practical application of the method, it must be capable of solving problems whose boundaries may be curved. In such cases, boundary nodes are not all exactly h units away from an inner node as is the case between adjacent inner nodes. The finite difference approximation (of the harmonic operator) at each inner node involves not only the variable value at that node and at the four surrounding nodes (above, below, left and right) but also the distance between these four surrounding nodes and the inner node - and at the boundaries these distances, quite likely, vary unpredictably. Compensation for the variation of these distances must be included in the finite difference solution. CLYDE represents the problem variable by a second degree polynomial in two variables, and employs a generalized irregular star in all directions for each inner node. In practice, one should not select such a coarse grid that more than (or even) two arms of the star are irregular (or less than h units in length). The generalized star permits (and automatically compensates for) a variation in length of any of the four arms radiating from a node. For no variation in any arm, the algorithm reduces exactly to the standard harmonic "computation stencil".

MATHEMATICAL MODEL

As the term implies, boundary value problems are those for which conditions are known at the boundaries. These conditions may be the value of the problem variable itself (temperature, for example), the normal gradient (or variable slope), or higher derivatives of the problem variable. For some problems, mixed boundary conditions may have to be specified: different conditions at different parts of the boundary. CLYDE solves those problems for which the problem variable, itself, is known at the boundary.

Given sets of equally spaced arguments and corresponding tables of function values the finite difference analyst may employ forward, central, and backward difference operators. CLYDE is based upon the central difference operators to approximate each differential operator in the equation.

The problem domain is overlaid with an appropriately selected grid. There are many shapes (and sizes) of overlaying cartesian and polar coordinate grids:

rectangular..
square..
equilateral - triangular..
equilangular - hexagonal..
oblique..

CLYDE uses a constant size (throughout the area of the problem) square grid for which the percentage errors are of the grid size squared (h^2). This grid or net consists of parallel vertical lines (spaced h units apart) and parallel horizontal lines (h units apart) which blanket the problem area from left-to-right and bottom-to-top.

CONSIDER THE GENERAL EXPRESSION:

$$\nabla^2 f = A\frac{\partial^2 f}{\partial \eta^2} + B\frac{\partial^2 f}{\partial \xi^2} + \frac{C}{\lambda}\frac{\partial f}{\partial \lambda} = D \quad ..EQ(1)$$

IN THE η, ξ, λ COORDINATE SYSTEM,

WHERE A, B, C, D ARE ARBITRARY CONSTANTS.

WHEN C=0, $\nabla^2 f$ REDUCES TO A TWO COORDINATE SYSTEM, SAY IN X AND Y:

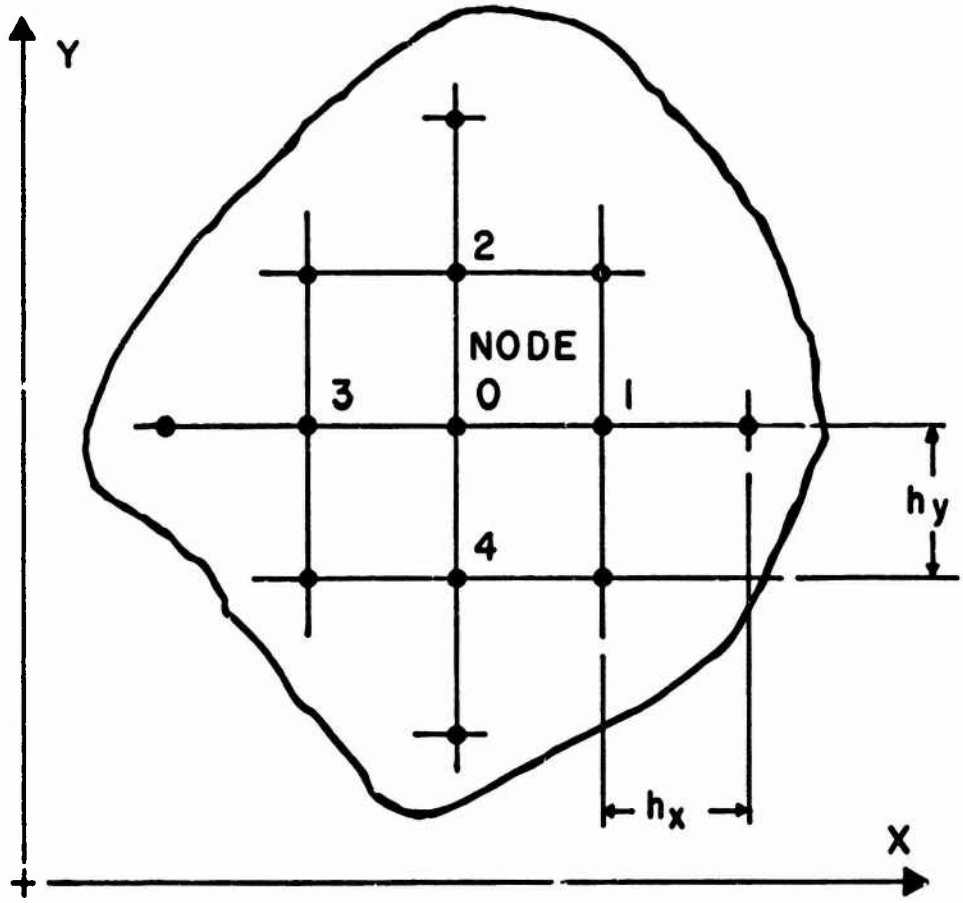
$$\nabla^2 f = A\frac{\partial^2 f}{\partial x^2} + B\frac{\partial^2 f}{\partial y^2} = D \quad ..EQ(2)$$

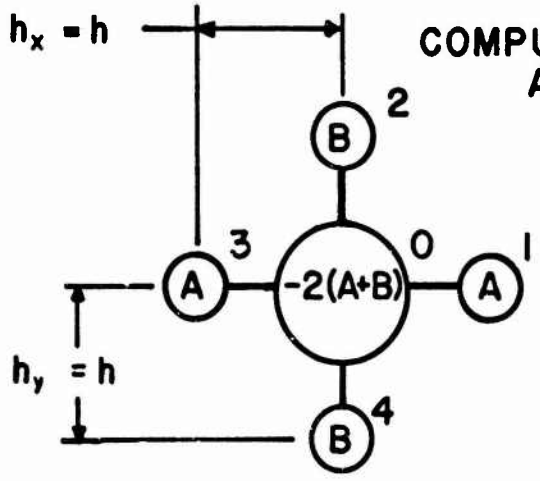
USING CENTRAL DIFFERENCES, THE FINITE DIFFERENCE APPROXIMATIONS TO THE PARTIAL DIFFERENTIAL OPERATORS, OF THE FUNCTION f, AT REPRESENTATIVE NODE 0 ARE:

$$\frac{\partial f}{\partial x} = \frac{1}{2h_x}(f_1 - f_3) \quad , \frac{\partial f}{\partial y} = \frac{1}{2h_y}(f_2 - f_4)$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{1}{h_x^2}(f_1 - 2f_0 + f_3)$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{1}{h_y^2}(f_2 - 2f_0 + f_4)$$





COMPUTATION STENCIL AT NODE 0

I. HARMONIC OPERATOR FOR SQUARE GRID

$$\nabla^2 f = A\frac{\partial^2 f}{\partial X^2} + B\frac{\partial^2 f}{\partial Y^2} = D$$

FOR A SQUARE GRID, $h_x = h_y = h$, AND THE HARMONIC OPERATOR $\nabla^2 f$ BECOMES:

$$h^2\nabla^2 f_0 = \left[A\,(f_1+f_3) + B\,(f_2+f_4) - (A+B)\,2f_0 \right] = h^2 D$$

..EQ (3)

SEE FIG. 1

THE NUMERICAL TREATMENT OF AN IRREGULAR STAR ($h_1 \neq h_2 \neq h_3 \neq h_4$) REPRESENTS THE FUNCTION f, NEAR THF REPRESENTATIVE NODE 0, BY A SECOND DEGREE POLYNOMIAL IN X AND Y

$$f(X,Y) = f_0 + a_1X + a_2Y + a_3X^2 + a_4Y^2 + a_5XY$$

EVALUATING THIS POLYNOMIAL AT THE NEIGHBORING NODES (1,2,3,4) PRODUCE THE FOLLOWING SET OF EQUATIONS:

$$f_1 = f_0 + a_1 h_1 + a_3 h_1^2$$

$$f_2 = f_0 + a_2 h_2 + a_4 h_2^2$$

$$f_3 = f_0 - a_1 h_3 + a_3 h_3^2$$

$$f_4 = f_0 - a_2 h_4 + a_4 h_4^2$$

WHICH ARE THEN SOLVED FOR a_3 AND a_4, WHICH ARE NECESSARY TO SATISFY THE HARMONIC OPERATOR $\nabla^2 f$, SINCE :

$$\frac{\partial f}{\partial x} = a_1 + 2a_3 X + a_5 Y, \quad \frac{\partial^2 f}{\partial x^2} = 2a_3$$

$$\frac{\partial f}{\partial Y} = a_2 + 2a_4 Y + a_5 X, \quad \frac{\partial^2 f}{\partial Y^2} = 2a_4$$

AND

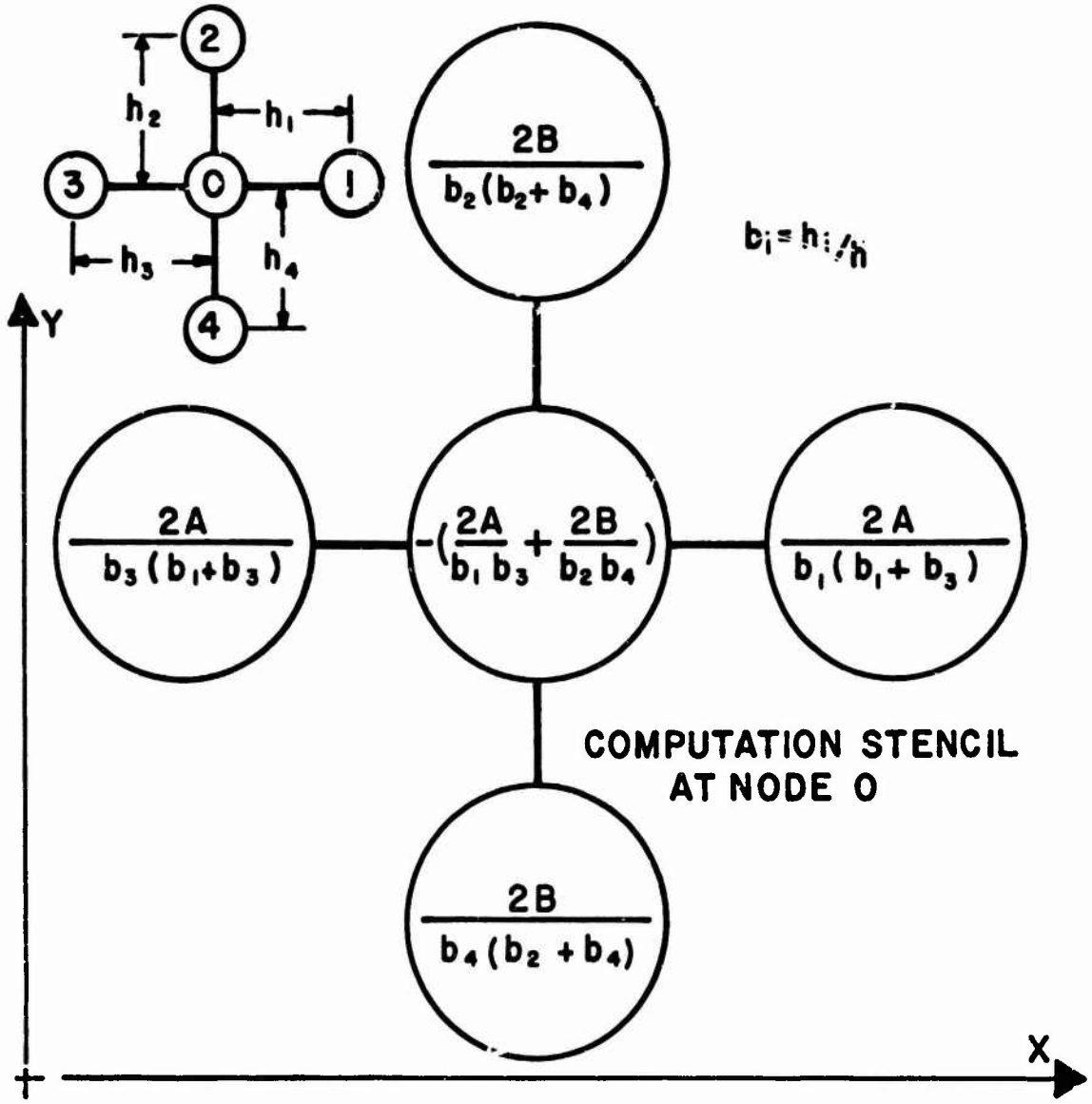
$$\nabla^2 f = A\,(2a_3) + B\,(2a_4)$$

PERFORMING THE NECESSARY ALGEBRAIC OPERATIONS, SUBSTITUTING RESULTS, COLLECTING TERMS, AND USING THE FOLLOWING RATIOS :

$$b_1 = \frac{h_1}{h} \qquad b_2 = \frac{h_2}{h}$$

$$b_3 = \frac{h_3}{h} \qquad b_4 = \frac{h_4}{h}$$

IRREGULAR STAR AT NODE 0
&
NEIGHBORING NODES (1, 2, 3, 4,)



2. HARMONIC OPERATOR FOR IRREGULAR GRID

$$\nabla^2 f = A\frac{\partial^2 f}{\partial X^2} + B\frac{\partial^2 f}{\partial Y^2} = D$$

THE HARMONIC OPERATOR BECOMES:

$$h^2\nabla^2 f_0 = \left[\frac{2A}{b_1(b_1+b_3)} f_1 + \frac{2B}{b_2(b_2+b_4)} f_2 + \right.$$
$$+ \frac{2A}{b_3(b_1+b_3)} f_3 + \frac{2B}{b_4(b_2+b_4)} f_4 +$$
$$\left. -\left(\frac{2A}{b_1 b_2} + \frac{2B}{b_2 b_4}\right) f_0\right] = h^2 D \quad ..EQ(4)$$

SEE FIG. 2

WHEN $C \neq 0$, $\nabla^2 f$ CAN BE APPLIED TO A (AXISYMMETRIC) CYLINDRICAL COORDINATE SYSTEM,

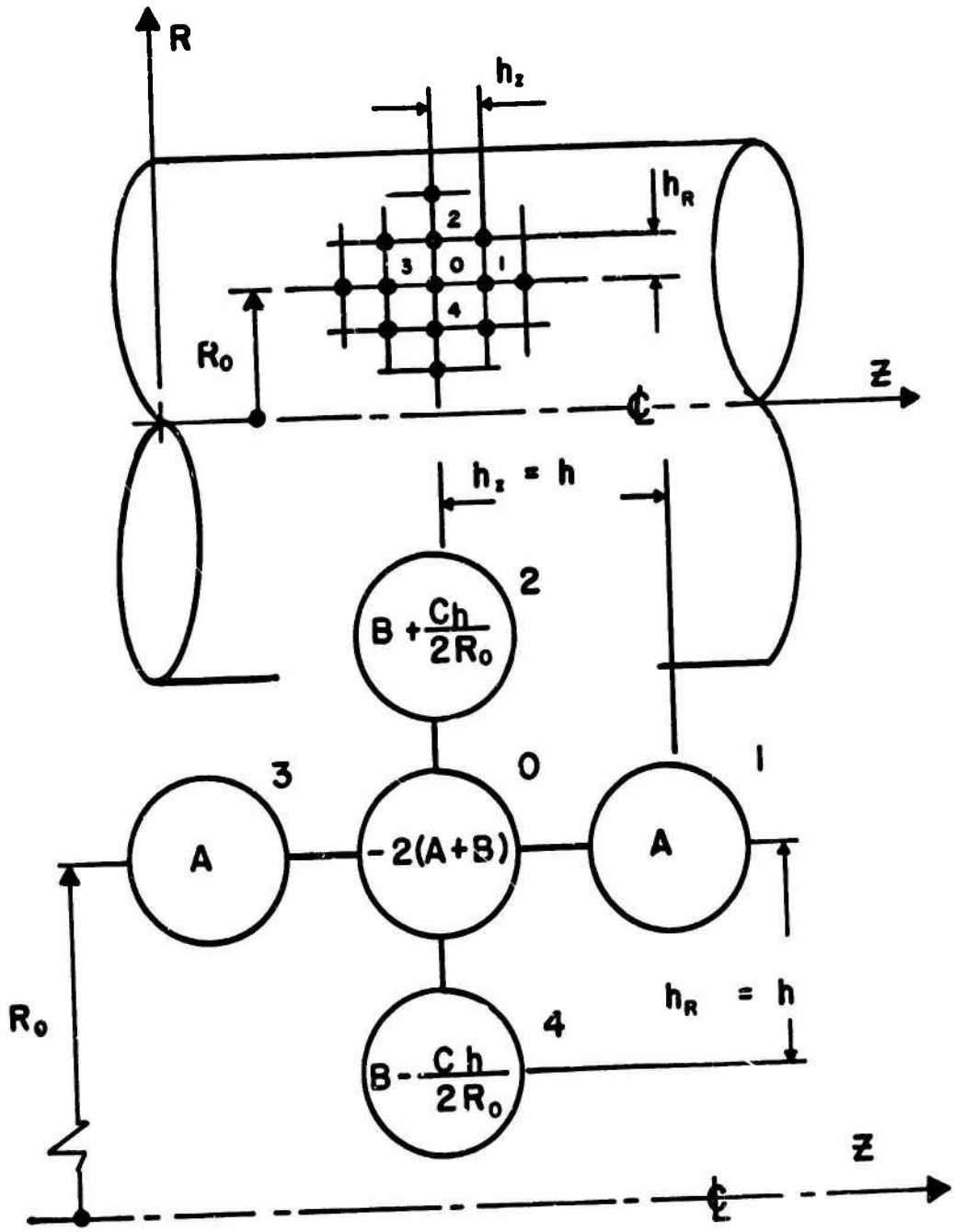
SAY IN R AND Z :

$$\nabla^2 f = A\frac{\partial^2 f}{\partial Z^2} + B\frac{\partial^2 f}{\partial R^2} + \frac{C}{R}\frac{\partial f}{\partial R} = D \quad ..EQ(5)$$

FOR A REGULAR STAR, THE HARMONIC OPERATOR BECOMES (IN A SIMILAR MANNER TO EQ (3)):

$$h^2\nabla^2 f_0 = \left[A(f_1+f_3) + B(f_2+f_4) + \frac{Ch}{2R_0}(f_2-f_4)\right.$$
$$\left. -(A+B)2f_0\right] = h^2 D \quad ..EQ(6)$$

SEE FIG. 3



3. HARMONIC OPERATOR FOR SQUARE GRID

$$\nabla^2 f = A\frac{\partial^2 f}{\partial z^2} + B\frac{\partial^2 f}{\partial R^2} + \frac{C}{R}\frac{\partial f}{\partial R} = D$$

FOR AN IRREGULAR STAR ($h_1 \neq h_2 \neq h_3 \neq h_4$)
THE HARMONIC OPERATOR BECOMES (IN A MANNER
SIMILAR TO EQ (4)):

$$h^2\nabla^2 f_0 = \left[\frac{2A}{b_1(b_1+b_3)}f_1 + \frac{2B}{b_2(b_2+b_4)}f_2 + \right.$$

$$+\frac{2A}{b_3(b_1+b_3)}f_3 + \frac{2B}{b_4(b_2+b_4)}f_4 +$$

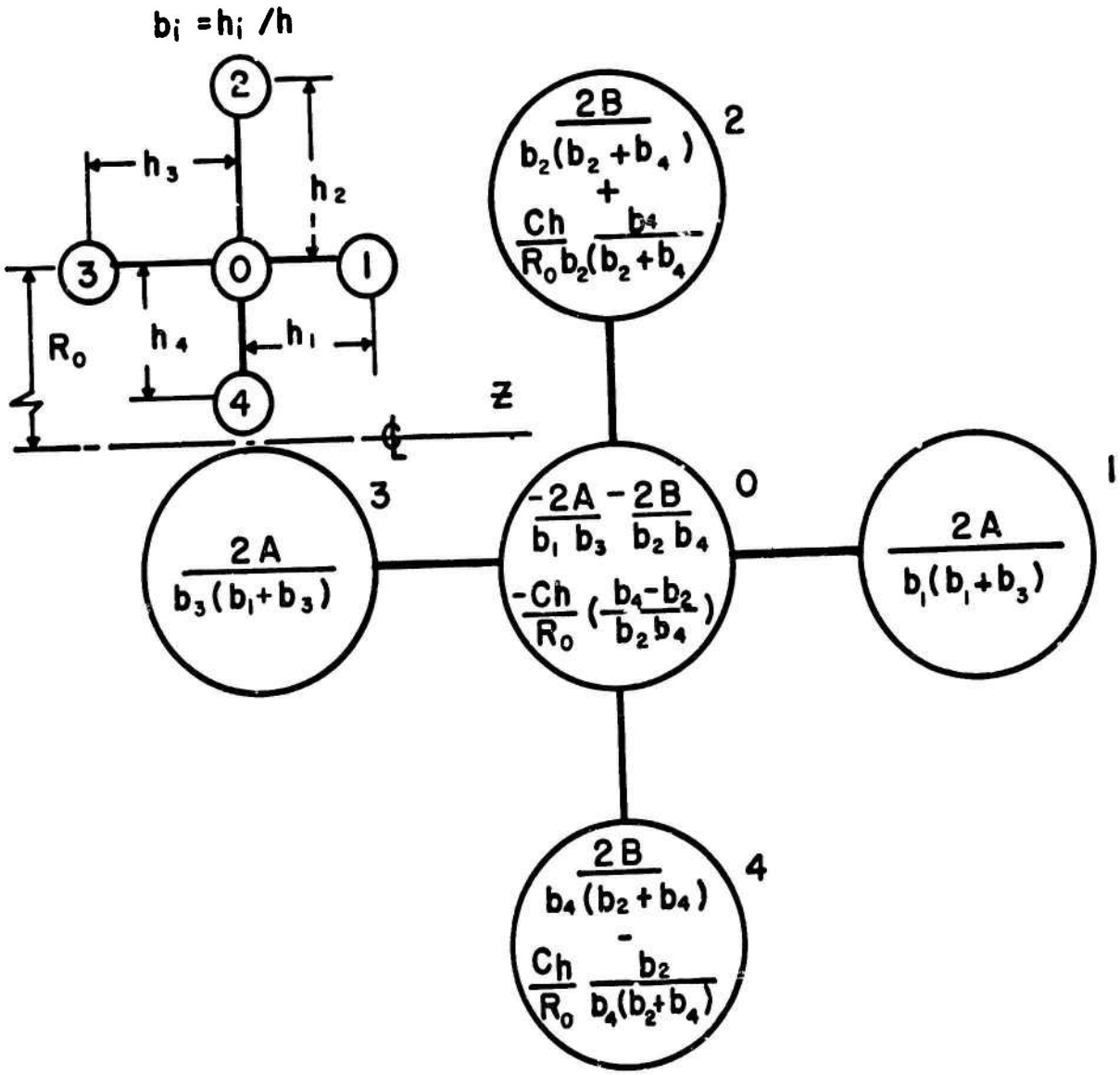
$$+\frac{Ch}{R_0}\left\{\frac{b_4}{b_2(b_2+b_4)}f_2 - \frac{b_2}{b_4(b_2+b_4)}f_4\right\} +$$

$$\left.-\left\{\frac{2A}{b_1 b_3} + \frac{2B}{b_2 b_4} - \frac{Ch}{R_0}\left(\frac{b_2-b_4}{b_2 b_4}\right)\right\}f_0\right]$$

$$= h^2 D \qquad \text{..EQ (7)}$$

SEE FIG. 4

EQS (4) AND (7) ARE EMPLOYED IN THE PROGRAMMED SOLUTIONS FOR CARTESIAN AND CYLINDRICAL COORDINATES, RESPECTIVELY.



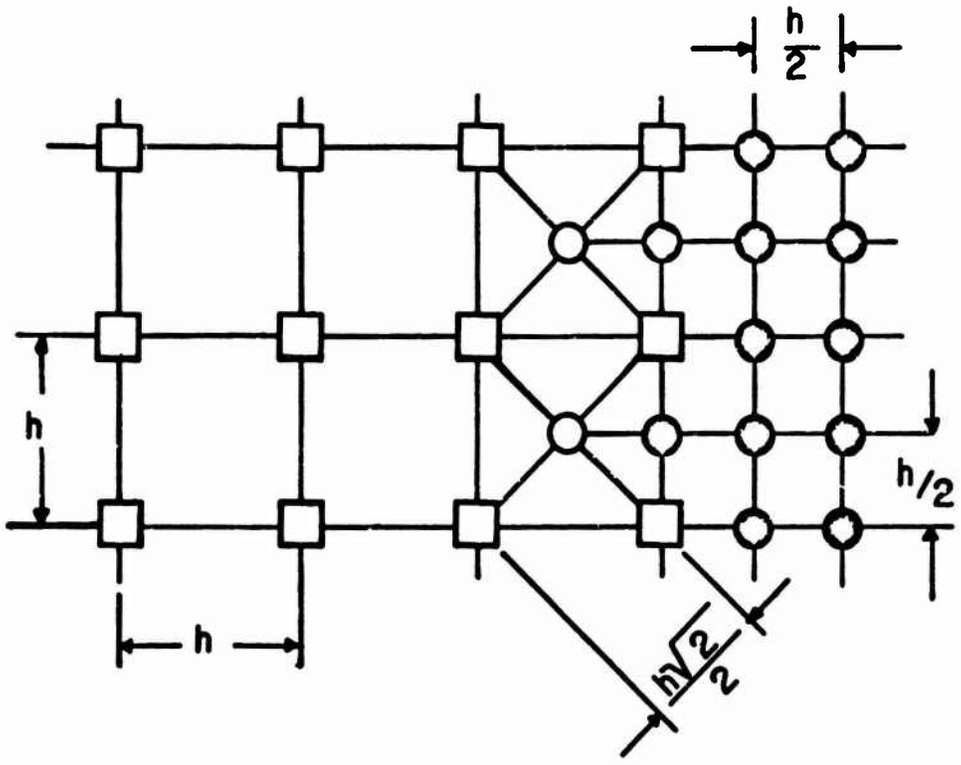
4. HARMONIC OPERATOR FOR IRREGULAR GRID

$$\nabla^2 f = A\frac{\partial^2 f}{\partial Z^2} + B\frac{\partial^2 f}{\partial R^2} + \frac{C}{R}\frac{\partial f}{\partial R} = D$$

GRADED MESH

THE MESH SIZE MAY BE REDUCED IN CRITICAL REGIONS, YIELDING A HIGHER RESOLUTION WHERE REQUIRED, WITHOUT THE COST OF AN EXCESSIVE NUMBER OF NODES OVER THE ENTIRE DOMAIN OF THE PROBLEM. THE FINER MESH IS TREATED WITH THE SAME EQUATIONS (EQS.(4) & (7) AS THE ORIGINAL, BUT WITH THE NEW SPACING, h.

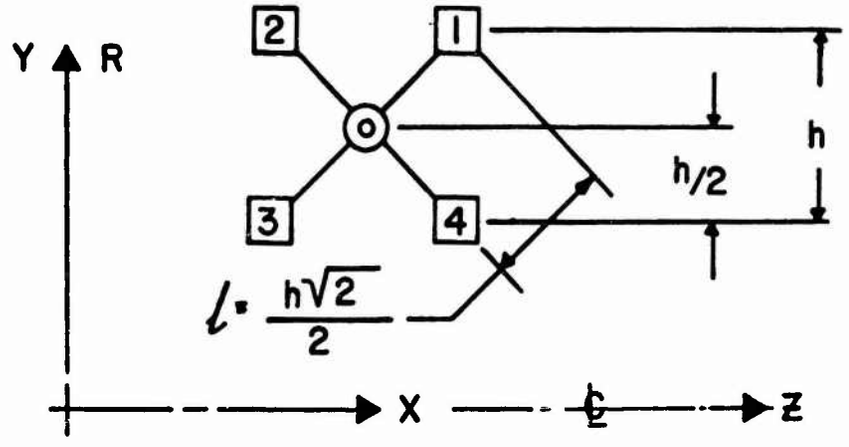
BETWEEN THE ORIGINAL (COARSE) AND NEW (FINER) MESH, HOWEVER, THERE EXISTS AN INTERMEDIATE MESH OR NET THAT REQUIRES SPECIAL TREATMENT. THIS INTERMEDIATE MESH WILL NOW BE CONSIDERED FOR BOTH CARTESIAN (EQ (4)) AND CYLINDRICAL (EQ (7)) COORDINATE SYSTEMS. NOTE THAT INTERMEDIATE MESH GRIDS ARE "SQUARE". THAT IS, ALL ARMS OF THE STAR ARE EQUAL ($h_1=h_2=h_3=h_4=h$).



□ ORIGINAL MESH (OR NET) NODE

○ FINER MESH NODE

○ INTERMEDIATE MESH NODE



USING ABOVE NOTATION FOR INTERMEDIATE NODES & USING "AVERAGING" DIFFERENCES:

$$\ell^2 \nabla^2 f_{x,y} \equiv A\frac{\partial^2 f}{\partial X^2} + B\frac{\partial^2 f}{\partial Y^2} = D\ell^2$$

BECOMES

$$\left(\frac{A+B}{2}\right)\left[f_1 + f_2 + f_3 + f_4 - 4f_0\right] = D\ell^2$$

$$= D\frac{h^2}{2} \qquad \text{..EQ(8)}$$

$$\ell^2 \nabla^2 f_{R,Z} \equiv A\frac{\partial^2 f}{\partial Z^2} + B\frac{\partial^2 f}{\partial R^2} + \frac{C}{\partial R}\frac{\partial f}{\partial R^2} = D\ell^2$$

BECOMES

$$\left(\frac{A+B}{2} + \frac{C\ell}{4R}\right)(f_1 + f_2) + \left(\frac{A+B}{2} - \frac{C\ell}{4R}\right)(f_3 + f_4)$$

$$-\left(\frac{A+B}{2}\right) 4f_0 = D\ell^2 = \frac{Dh^2}{2} \qquad \text{..EQ(9)}$$

WHERE $\frac{C\ell}{4R}$ IS $\frac{Ch}{8R}\sqrt{2}$

THE BIHARMONIC OPERATOR

$$\nabla^4 W = \frac{\partial^4 W}{\partial x^4} + 2\frac{\partial^4 W}{\partial x^2 \partial Y^2} + \frac{\partial^4 W}{\partial Y^4} = \frac{q}{D} \qquad \cdot\cdot EQ\ (10)$$

CAN BE REPLACED BY TWO SECOND ORDER EQUATIONS

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial Y^2}\right)\left(\frac{\partial^2 W}{\partial x^2} + \frac{\partial^2 W}{\partial Y^2}\right) = \frac{q}{D} \qquad \cdot\cdot EQ\ (11)$$

SINCE

$$M_x = -D\left(\frac{\partial^2 W}{\partial x^2} + \mu \frac{\partial^2 W}{\partial Y^2}\right) \quad \text{AND}$$

$$M_y = -D\left(\frac{\partial^2 W}{\partial Y^2} + \mu \frac{\partial^2 W}{\partial x^2}\right)$$

$$M_x + M_y = -D(1+\mu)\left(\frac{\partial^2 W}{\partial x^2} + \frac{\partial^2 W}{\partial Y^2}\right)$$

INTRODUCING A NEW NOTATION

$$M = \frac{M_x + M_y}{1+\mu} = -D\left(\frac{\partial^2 W}{\partial x^2} + \frac{\partial^2 W}{\partial Y^2}\right)$$

EQ (11) MAY BE REPRESENTED BY

$$\frac{\partial^2 M}{\partial x^2} + \frac{\partial^2 M}{\partial Y^2} = -q \qquad \cdot\cdot EQ\ (12a)$$

$$\frac{\partial^2 W}{\partial x^2} + \frac{\partial^2 W}{\partial Y^2} = -\frac{M}{D} \qquad \cdot\cdot EQ\ (12b)$$

HYBRID COMPUTER SOLUTION TECHNIQUES FOR LAPLACE'S EQUATION

J. Thomas Broach and Robert M. McKechnie III
U. S. Army Mobility Equipment Research and Development Center
Fort Belvoir, Virginia

ABSTRACT. The techniques for hybrid computer graphics solution of Laplace's equation are discussed. This describes two approaches, problem setup requirements, and compares the hybrid solution of an electromagnetic field problem to the exact solution. The hybrid system being used is an AD-4 analog computer/PDP-15 digital minicomputer coupled to a Tektronix 4010 graphics system.

This effort is the forerunner to the development of an interactive graphics programming language for the hybrid computer solution of partial differential equations.

The discussion will present the general philosophy and details of the solution techniques being used for hybrid computer solution of Laplace's equations with an example being used for method demonstration. This approach will eventually lead to a comparison between the pure digital and hybrid solutions and should verify the general feeling that the hybrid computer can provide a faster and lower cost solution.

1. INTRODUCTION. The Electrical Equipment Division is involved in the solution of partial differential equations for heat transfer and magnetic flux in electric and electronic machinery. The subject of this paper is the solution of an electromagnetic field problem, similar to those encountered in the design of electric machines. This problem is being solved on the MERDC Computer Aided Design and Engineering (CAD-E) facility located in the Electrical Equipment Division.

2. SYSTEM DESCRIPTION. The CAD-E facility presently consists of a graphics terminal, hybrid computer, and communications to outside computers as shown in Figure 1.

The interactive graphics terminal consists of a Varian 620 minicomputer processor with 32K of core (16 bit words) memory, four magnetic tape devices (COI link tapes), high speed (200 characters/second) paper tape I/O, 400 line per minute Vogue line printer, Infoton alphanumeric CRT I/O, ARDS 100A graphics unit with joystick, Tektronix 4010 graphics unit with hardcopy unit, KSR 35 teletype and a highspeed disk.

The hybrid computer is an AD4/PDP15 hybrid system. It consists of an Applied Dynamics AD4 analog processor (Figure 2), which has two quadrants of integrators, amplifiers, servo-set pots (1 quadrant), hand set pots, digital coefficient units (DCU)

(1 quadrant), multipliers, DAC's digital logic large screen oscilloscope, and XY plotter. Autopatch hardware has been installed in one quadrant, also the PDP15 digital processor (Figure 3) consists of 16K of core (18 bit words) memory, 3 DEC (magnetic) tape units, a disk, and a KSR-33 teletype. The Tektronix 4010 graphics unit (Figure 4) can be connected directly to the PDP15 digital processor with 12,000 Baud link. The hybrid system and the graphics terminal are connected through a 9600 Baud link between the Varian 620 and the PDP15 digital processors.

The communications link ties this facility to outside computers. One such tie is accomplished through a MODEM to outside contracts, commercial time-sharing computers, and the MERDC CDC 6600 digital computer facility. At present 300 Baud is the maximum rate in this link but a 2000 Baud rate is planned for the tie to the MERDC 6600 digital computer. This link also contains an ASR 33 teletype, TSP plotter controller with Tektronix 601 storage CRT and XY plotter.

3. PROBLEM DESCRIPTION. The problem to be solved is an application of Laplace's equation (Figure 5) for the solution of a magnetic field involving rectangular boundaries. This is the first step in developing a method of solving for magnetic fields in complicated machine geometries as shown in Figure 6. The corresponding hardware is shown in Figure 7. Since Laplace's equation has derivatives with respect to more than one independent variable, we usually convert it to a different form. On the digital computer, this results in a large set of equations which require a lot of computer time and memory. For the analog, the solutions require a lot of computing equipment. Through hybrid computing techniques, we retain the convenient man-machine interface with the digital computer, take advantage of the integration capability of the analog computer, and use the digital computer to control the analog allowing for equipment reduction.

During the early 1960's, a lot of work was accomplished for solution of partial differential equations on analog computers. With the expected use of hybrid computers, the emphasis was shifted to utilization of hybrid computers. However, the efforts since then have been small with little to show but theory. In the digital area, work has progressed, mainly due to the easier man-machine interface and through the efforts of universities and the large computer companies.

4. PROBLEM GEOMETRY. Now consider a rectangular geometry in which the potential on all four boundaries is defined, as shown by Figure 8. As shown, the potential is zero on three boundaries and is defined as a function of X, $\Psi(X) = \text{Sin}\frac{\pi X}{a}$, on the other boundary. For this type of problem, there are three common techniques of solution: (1) separation of variables, (2) finite difference, (3) Monte Carlo. Generally, we will use the finite difference technique because it can easily handle non-linearities. For a digital

solution, one reduces the partial differential equation to a set of algebraic equations using the finite difference technique. This means that iterative techniques must be employed to obtain solutions. For the hybrid, one obtains a set of ordinary differential equations using the finite difference technique.

5. SOLUTION TECHNIQUE. Normally the digital finite difference solution uses a two dimensional grid; however, since the analog computer solves continuously in one dimension, it uses a one dimensional grid as shown in Figure 9. For this problem we have taken advantage of the problem symmetry (normally, this is true for all electric machine problems) to provide additional grid lines and associated solutions. In this manner, we have judiciously chosen a non-linear grid spacing which provides a large amount of data with minimum of computation equipment.

One important factor is that we can use non-linear spacing and retain considerable accuracy. For this problem we need only four grid stations (excluding the two boundaries) Three stations are held in fixed locations (X = .167, .333, .5) while the fourth is moved in predefined increments in the space to the right of center, thus with symmetry obtaining a large number of solutions. The differential equations for each station are ordinary 2nd order differential equations with a finite difference term, as shown in Figure 10. Mechanization on the analog subsection of the hybrid is simple and requires no more than 8 integrators, as shown in Figure 11. The digital subsection of the hybrid controls the scanning process described above.

The problem control Flow Chart, Figure 12, shows how the hybrid computer is used to solve the problem. The user enters data for the problem at the Tektronix 4010 Graphics Terminal. From this point the digital computer subsection scales the problem, sets the pots and controls the analog subsection operate and hold functions. The analog subsection is used as a computation module to provide differential equation solutions at each grid station. The digital samples the analog output, operates on the data, and provides data to the Tektronix 4010 for a graphics solution. A listing of the data is also available.

Figure 13 is a picture of the analog patchboard required to solve this problem. The solution results are shown in Figure 14 in the form of an equipotential plot on the Tektronix 4010. The grid and labeling are performed as a substask on the digital computer.

6. CONCLUSIONS. This method of solving partial differential equations is being used on two special classes of problems: electric machinery and semiconductor heat transfer. The simplicity of this approach has made it easy to mechanize; however, two questions come to mind. First, is it accurate enough to be useful. For one problem, we utilized the exact analytical solution and compared the results to those obtained from

the grid solution on the hybrid. The close agreement is shown by Figure 15, proving that the accuracy is adequate. Secondly, do we save time? Using a comparison to the digital computer finite-difference equation solution, the digital took 3 hours and our hybrid took 15 minutes to provide the equipotential plots. This time comparison will be made in more depth as we increase problem complexity.

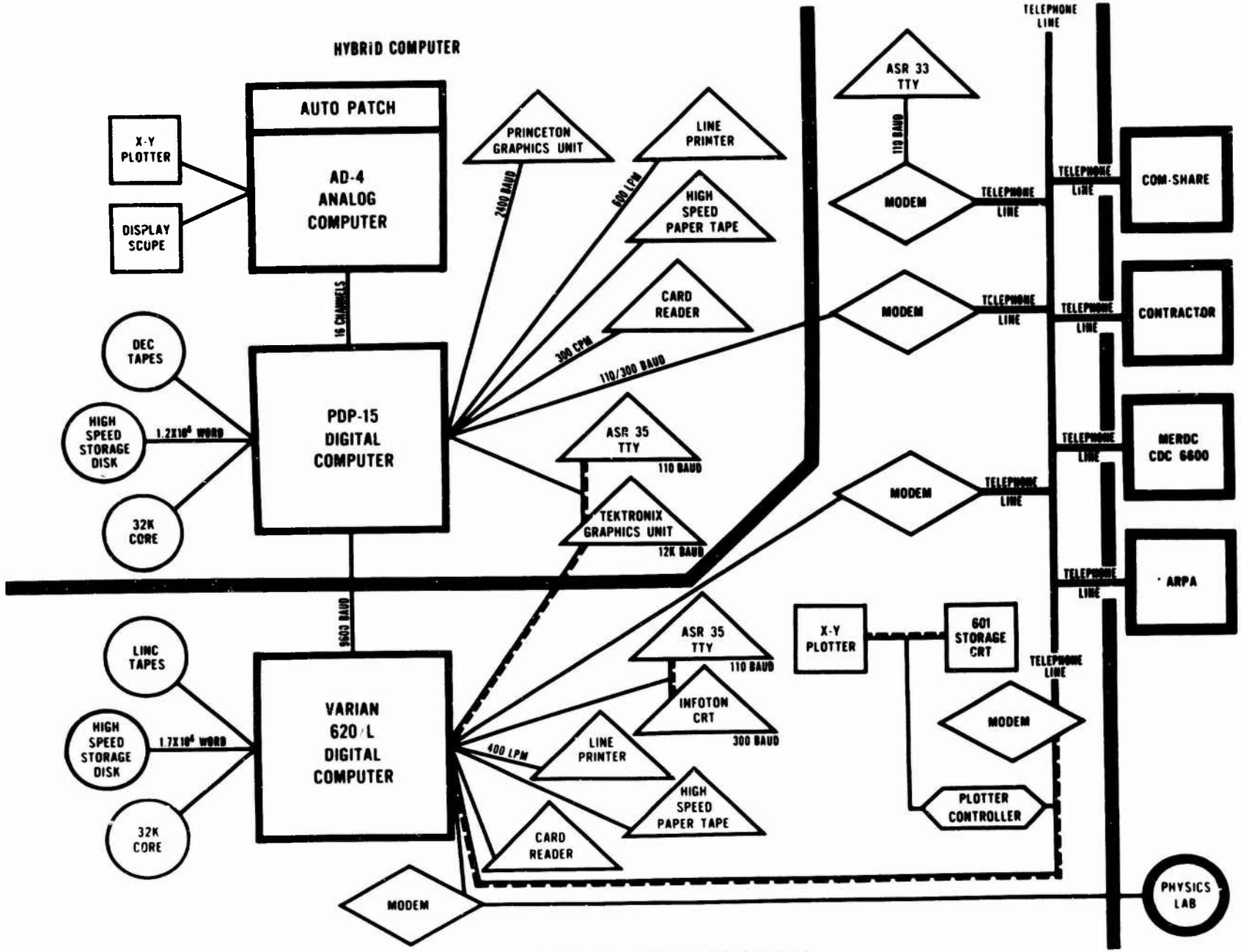
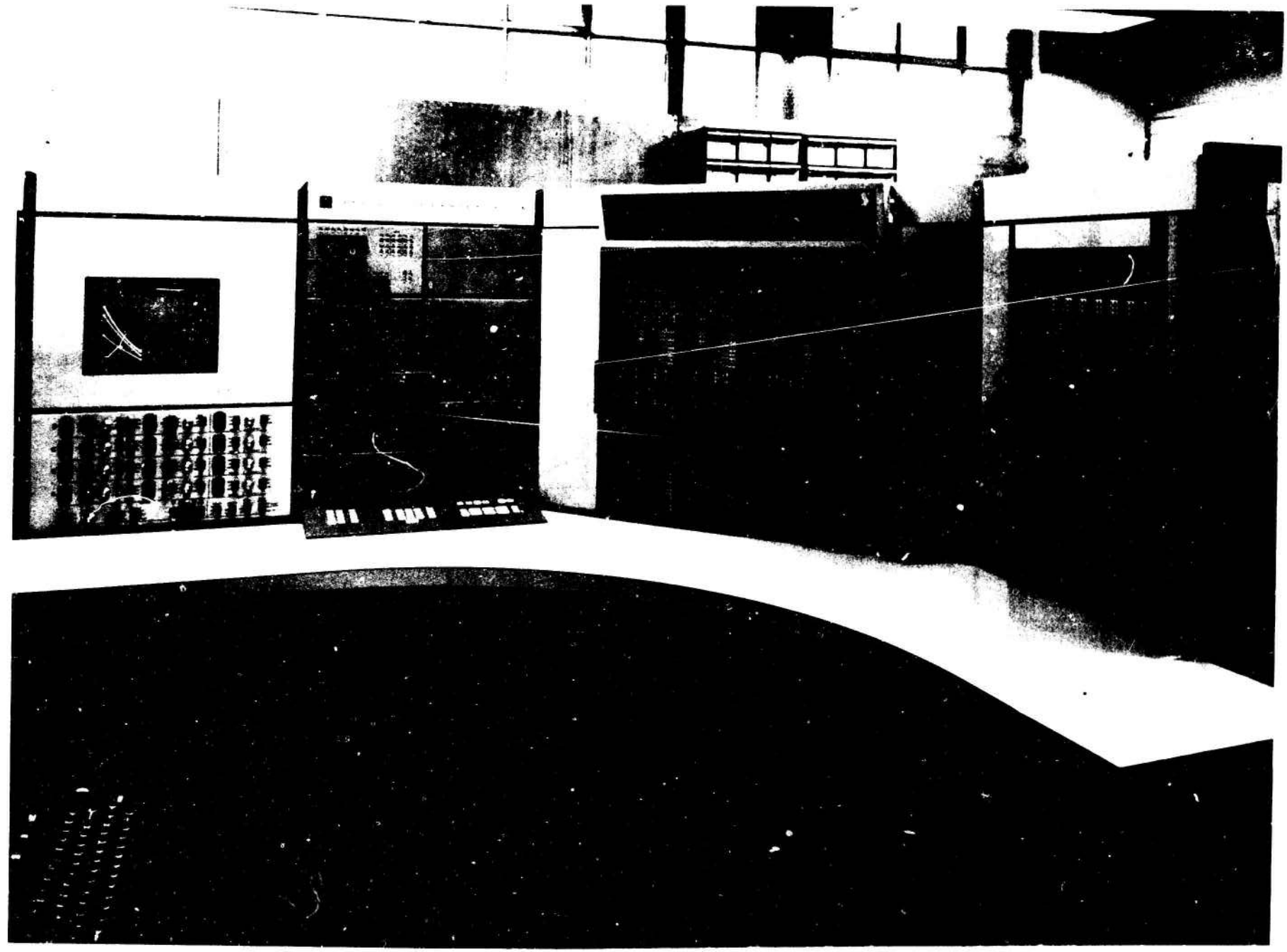


FIG. 1. CAD-E FACILITY.



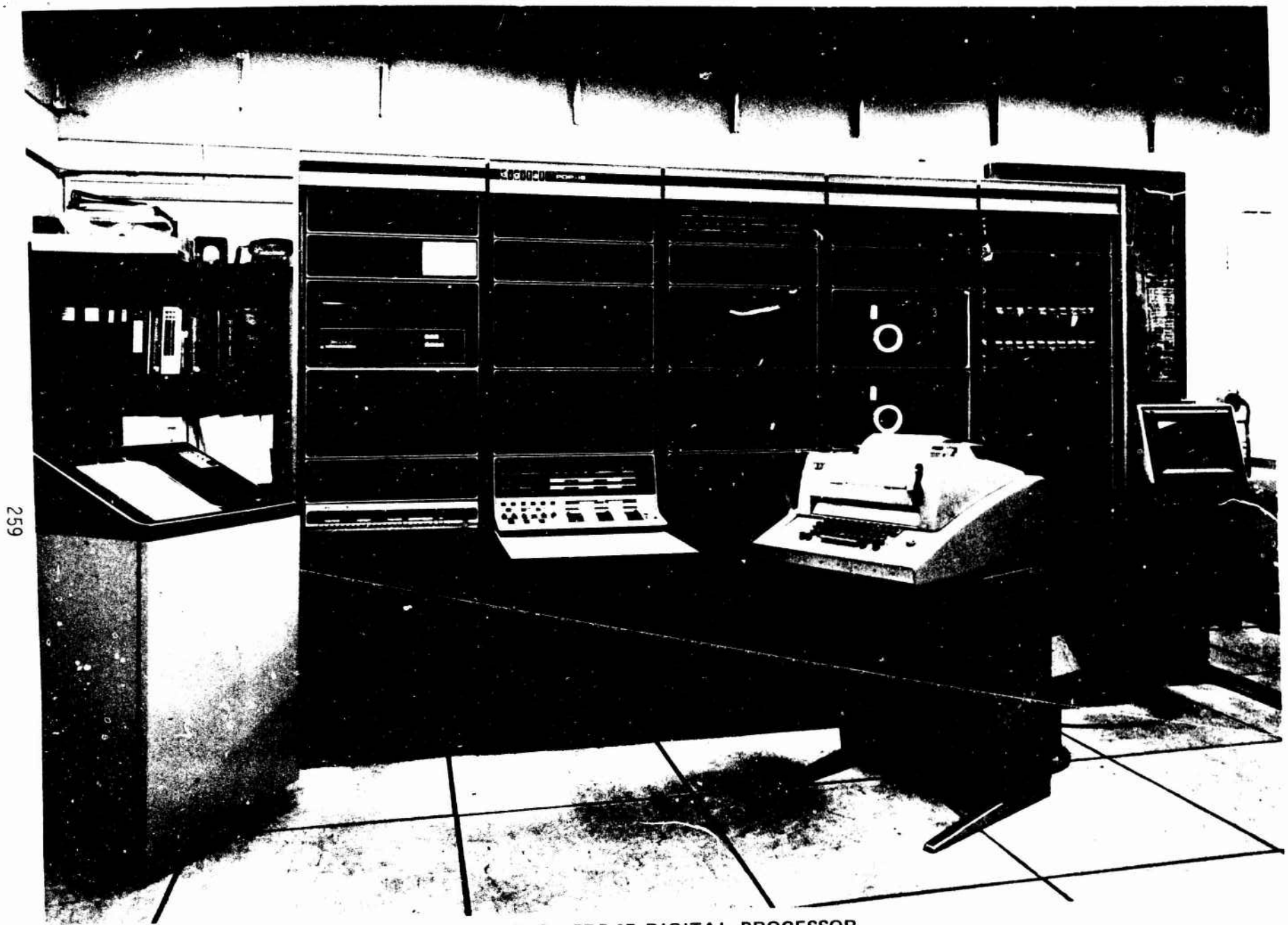
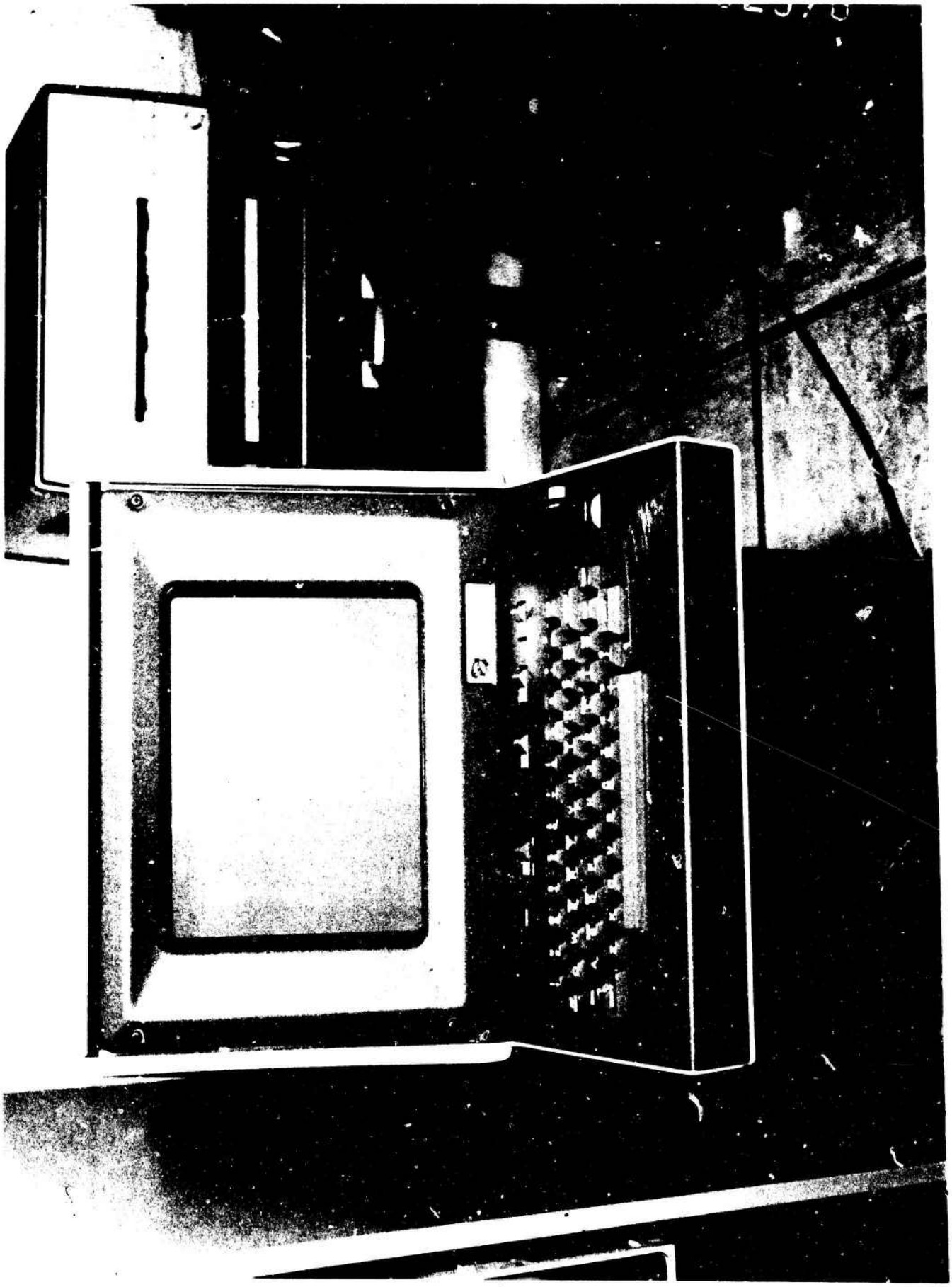


FIG. 3. PDP-15 DIGITAL PROCESSOR.



LAPLACES EQUATION

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0$$

FIG. 5. LAPLACE'S EQUATION.

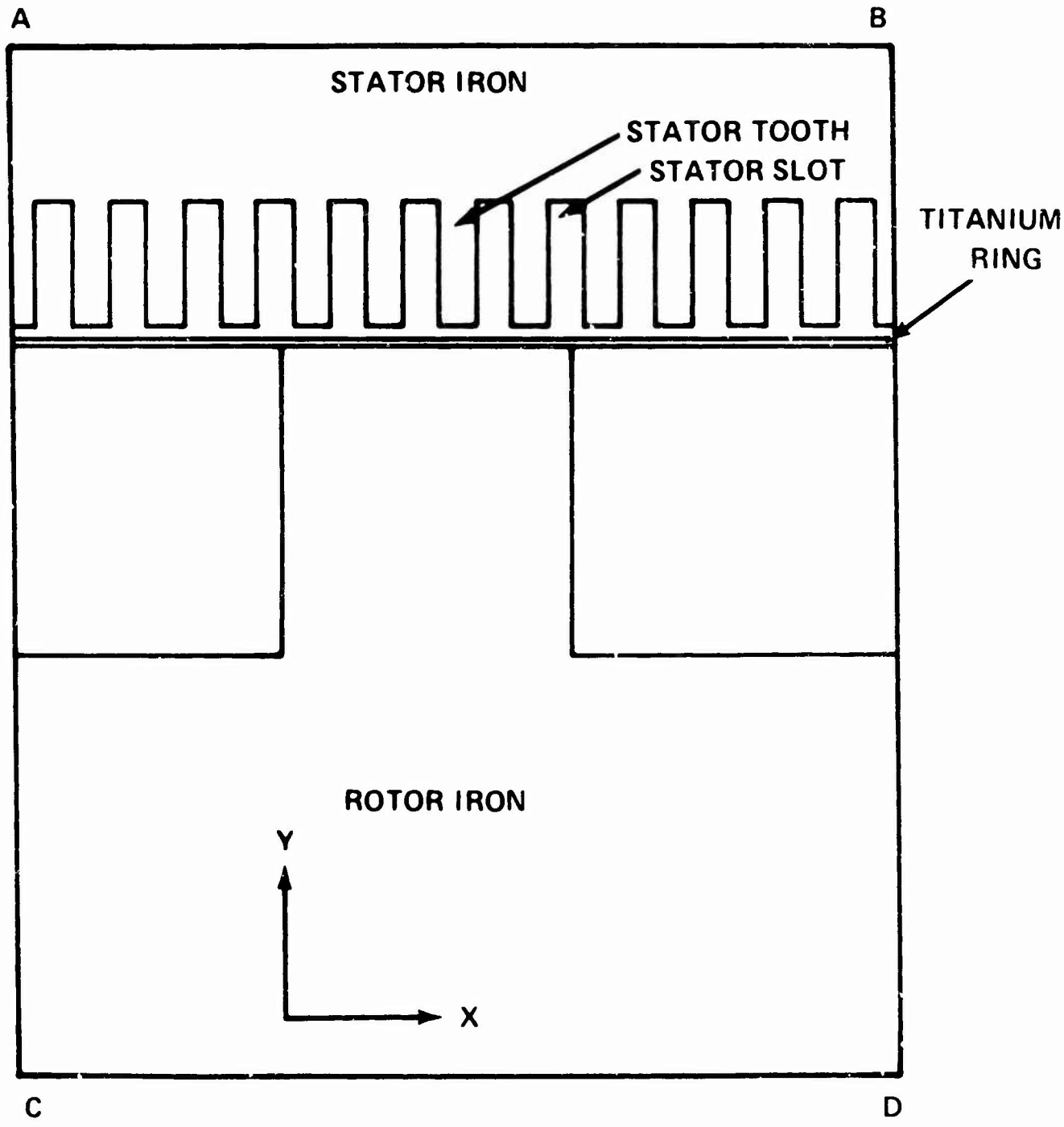
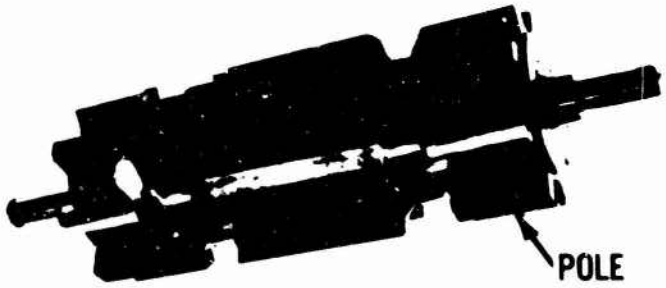
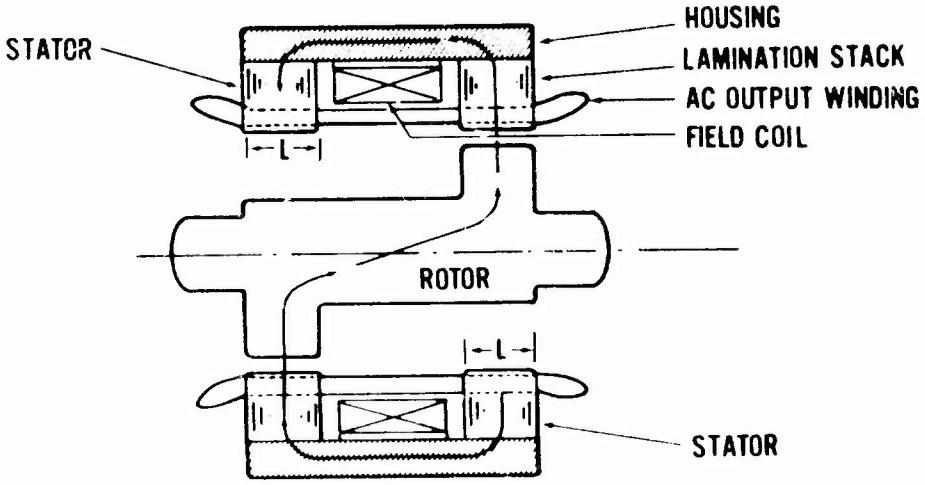


FIG. 6. TYPICAL ELECTROMAGNETIC MACHINE GEOMETRY.



SOLID ROTOR



FLUX PATH

TWO-SECTION HOMOPOLAR INDUCTOR ALTERNATOR

FIG. 7. ELECTROMAGNETIC MACHINE HARDWARE.

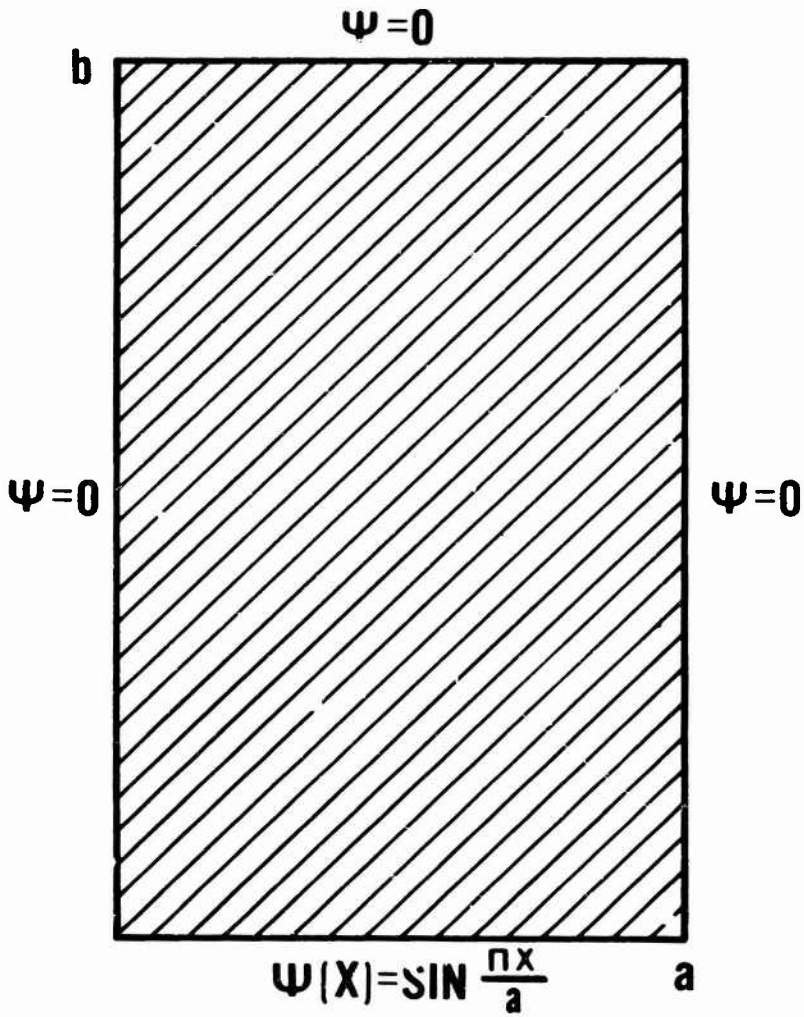


FIG. 8. PROBLEM GEOMETRY.

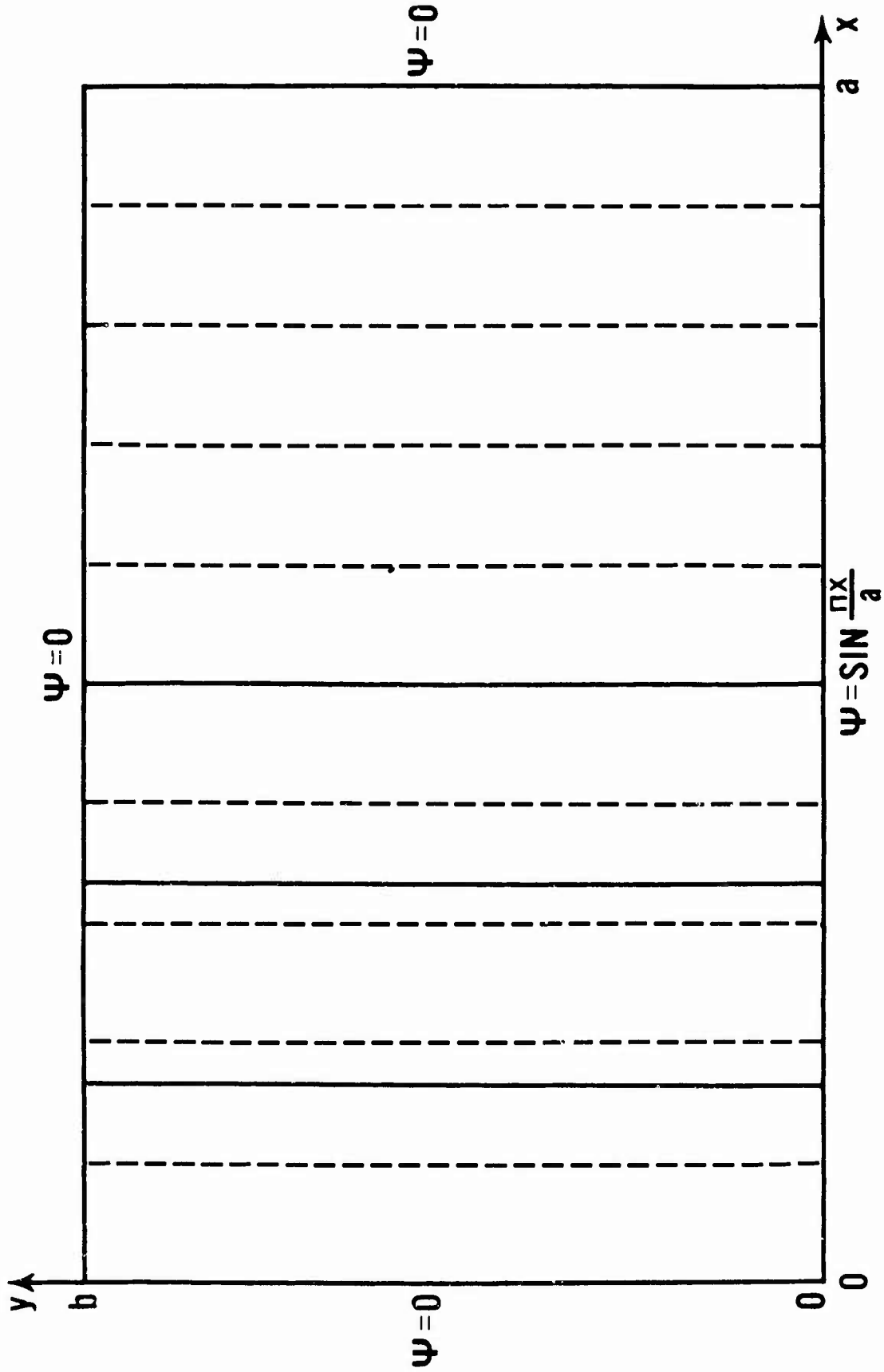


FIG. 9. ONE DIMENSIONAL GRID.

BASIC FINITE DIFFERENCE SCHEME FOR HYBRID COMPUTER

A CHANGE TO AN ORDINARY 2nd ORDER DIFFERENTIAL EQUATION AT EACH X-STATION

$$\ddot{\psi}_1 = \frac{1}{\Delta X_{11}} \left[\phi_{1\,2} - \phi_{3\,2} \right]$$

$$\ddot{\psi}_2 = \frac{1}{\Delta X_{21}} \left[\phi_{3\,2} - \phi_{5\,2} \right]$$

$$\ddot{\psi}_3 = \frac{1}{\Delta X_{31}} \left[\phi_{5\,2} - \phi_{7\,2} \right]$$

$$\ddot{\psi}_4 = \left[\frac{1}{\Delta X_{41}} \right] \left[\phi_{7\,2} - \phi_{9\,2} \right]$$

WHERE-

$$\phi_{1\,2} = \left[\frac{1}{\Delta X_{12}} \right] \left[\psi_1 - \psi \right]$$

$$\phi_{3\,2} = \left[\frac{1}{\Delta X_{22}} \right] \left[\psi_2 - \psi_1 \right]$$

$$\phi_{5\,2} = \left[\frac{1}{\Delta X_{32}} \right] \left[\psi_3 - \psi_2 \right]$$

$$\phi_{7\,2} = \left[\frac{1}{\Delta X_{42}} \right] \left[\psi_4 - \psi_3 \right]$$

$$\phi_{9\,2} = \left[\frac{1}{\Delta X_{52}} \right] \left[\psi_5 - \psi_4 \right]$$

FIG. 10. FINITE DIFFERENCE EQUATIONS.

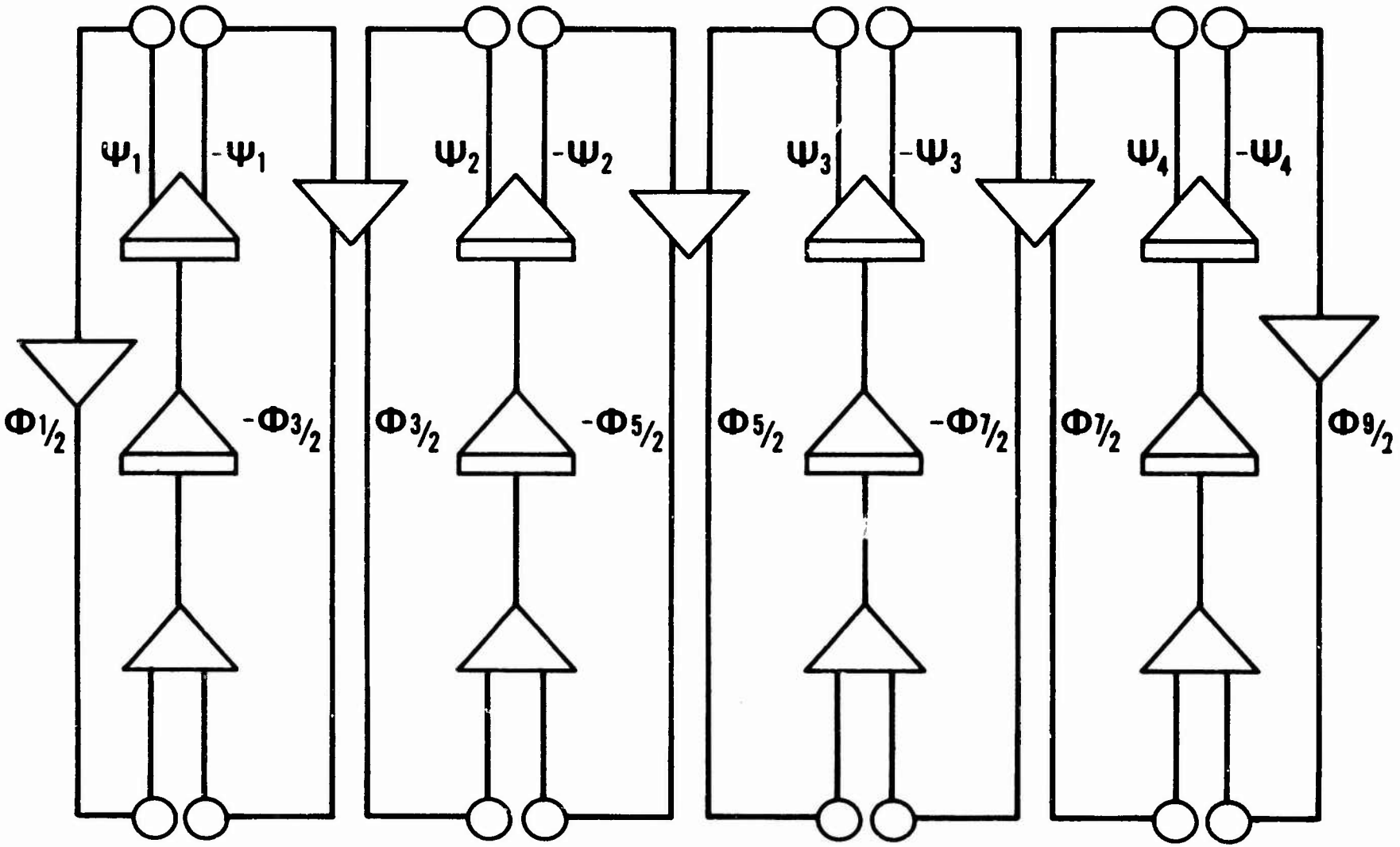


FIG. 11. ANALOG PATCHING DIAGRAM.

PROBLEM CONTROL FLOW CHART

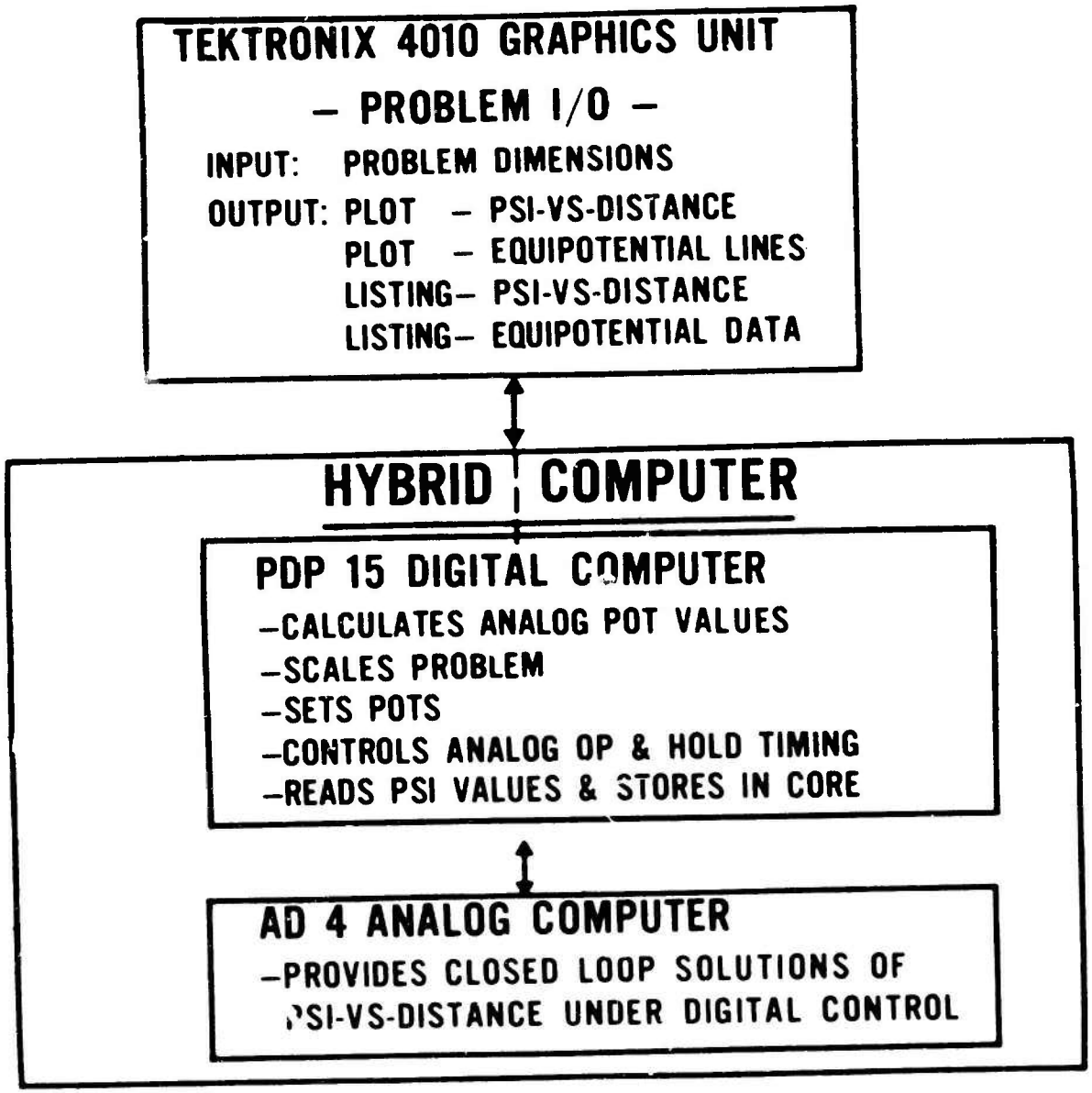


FIG. 12. PROBLEM CONTROL FLOW CHART.

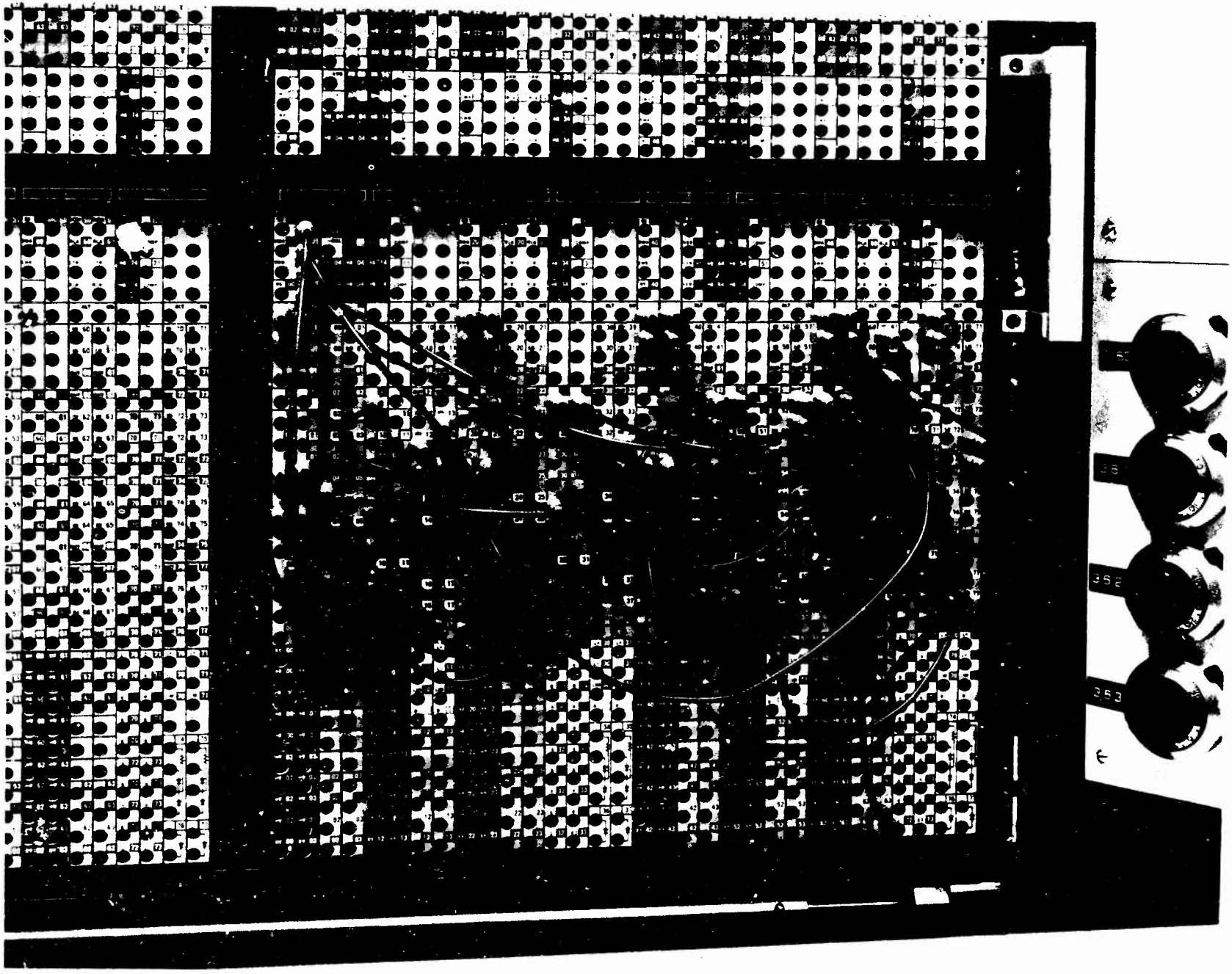
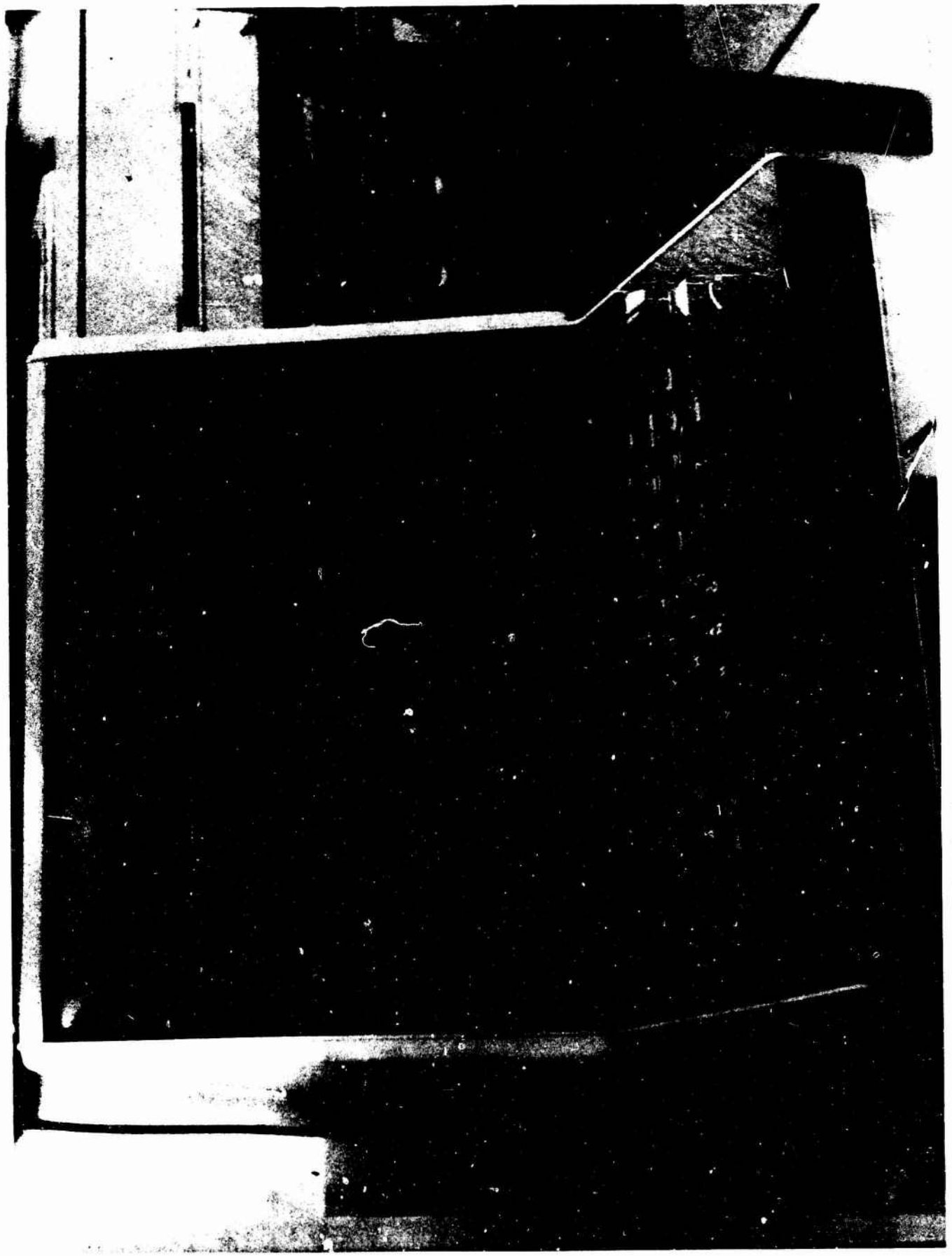


FIG. 13. ANALOG PATCHBOARD CONNECTIONS.



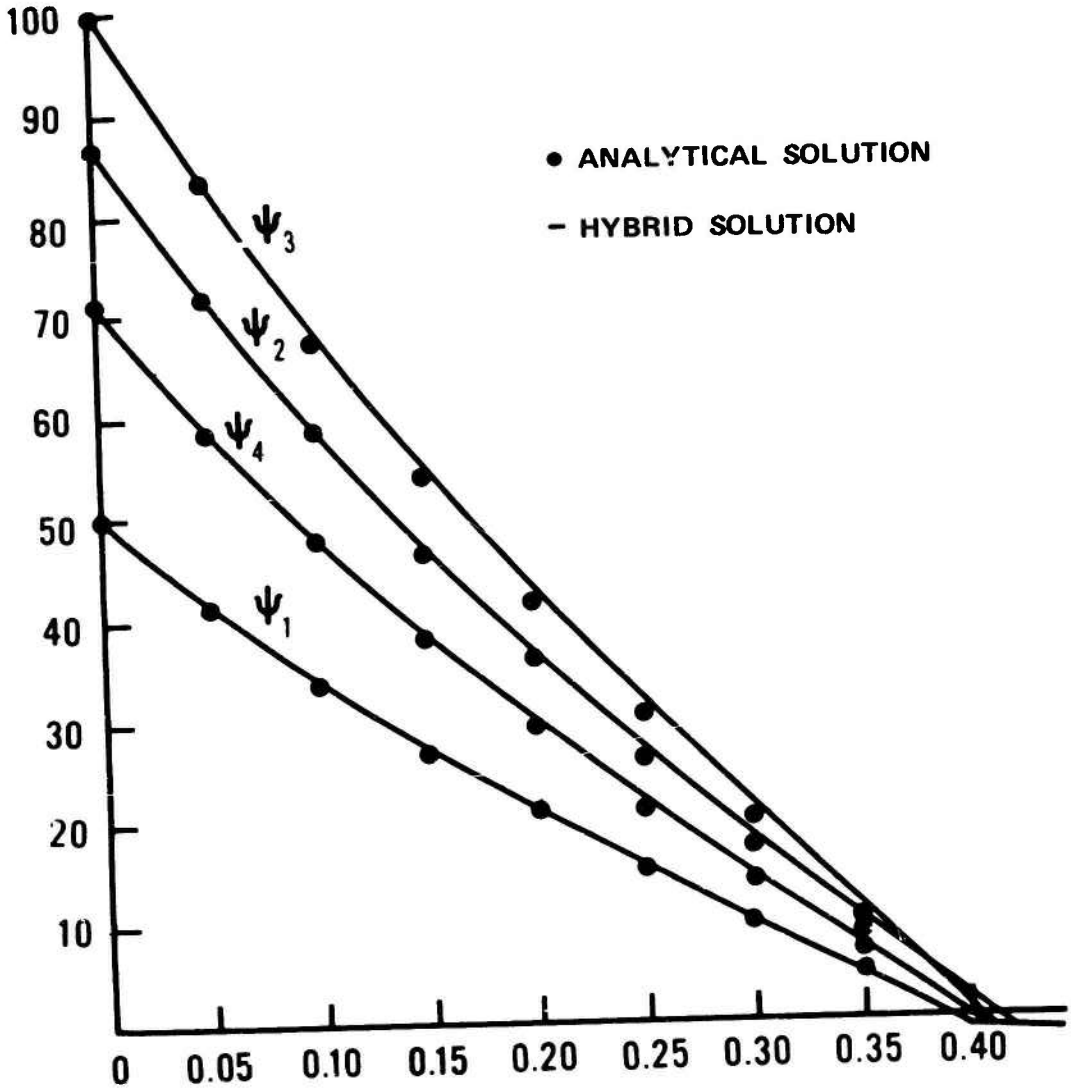


FIG. 15. COMPARISON OF ANALYTICAL AND HYBRID SOLUTIONS.

ANALYSIS PROCEDURE FOR OPTIMIZING HELIUM REFRIGERATION CYCLES

RUSSELL EATON, III
and
LARRY AMSTUTZ

U.S. ARMY MOBILITY EQUIPMENT RESEARCH AND DEVELOPMENT CENTER
FORT BELVOIR, VIRGINIA

ABSTRACT. An analysis procedure for optimizing helium refrigeration cycles has been developed as part of the Army's cryogenic refrigerator and CAD-E programs. The analysis procedure has been converted into a digital computer program consisting of subroutines which represent the basic components, such as heat exchangers and expansion engines, of a refrigerator. A cycle is built up by successively calling these "building block" subroutines starting at the lowest temperature and pressure and proceeding to ambient conditions. The minimum amount of information needed to specify each cycle was used as input, and the computer program supplied the resulting thermodynamic state points and component characteristics. Cycles were optimized for minimum input power by systematic trial and error method. Because the program is highly interactive, the minimum input power for any particular cycle could be found quickly. A graphical subroutine enabled the results of an optimized cycle to be displayed on Tektronix 4010 graphics system (Fig 1) as a schematic drawing of the refrigerator cycle with the state points and the important component characteristics labelled.

1. INTRODUCTION. The authors became interested in designing optimized thermodynamic cycles for cryogenic refrigerators as part of an Army program to develop reliable cryogenic refrigerators suitable for integrating with electrical power devices using superconductors. The techniques for designing and optimizing a thermodynamic cycle are straight forward, but these techniques lead to extremely tedious calculations, making it practically impossible to optimize cycles by hand. Typical relationships to be evaluated are conservation of energy, conservation of mass, and component efficiencies. The preceding relationships depend upon the following thermodynamic functions

$$
\begin{aligned}
&\text{Enthaply:} \quad H=h\ (P,T,\dot{m}) \\
&\text{"Inverted" Form:} \quad T=t\ (H,P,\dot{m}) \\
&\text{Enthropy:} \quad S=s\ (P,T,\dot{m}) \\
&\text{"Inverted" Form:} \quad T=t\ (S,P,m)
\end{aligned}
\tag{1}
$$

where P is pressure, T temperature, and $\dot{m}$ mass flow rate. The thermodynamic functions, enthalpy and entropy, are well tabulated, but in any cycle calculation, specific values of these functions usually have to be obtained by interpolation of either or both the thermodynamic variables, pressure and temperature.

At least three significant figures are needed for these interpolation calculations. In actual practice, the authors found that it took about one manday for a typical cycle to be analyzed excluding any optimization calculations.

The need to optimize refrigeration cycles with respect to design parameters, such as pressure ratio or heat exchanger effectiveness, provided the impetus for developing a computer aided refrigeration cycle analysis code. Additional benefits from using a computer approach are the saving in engineering time and the increased productivity, i.e., many more cycles options can be considered in a given time. Of course, the major point is that the emphasis of the cryogenic design engineer is shifted from arithmetic to engineering.

2 APPROACH. The authors have written an interactive program to analyze Claude cycle refrigerators. A schematic diagram of a Claude cycle with two expansion engines and a total thermal load consisting of a fixed load at the lowest temperature stage (4.5K) and the load due to a pair of 6000A current leads is shown in Fig 2 The thermal loads shown represent a typical load profile for some cryogenic electrical power devices. The analysis of such a cycle can be handled conveniently by starting from the lowest temperature (4.5K), adding components and loads until ambient

temperature is reached. During the analysis, design decisions are made by selecting component characteristics, such as heat exchanger effectiveness and expansion engine efficiency, or temperature stages of the cycle. The user may select one of several parameters available. For example, he may choose either a stage temperature or a heat exchanger effectiveness. The analysis proceeds with the engineer supplying the minimum number of inputs necessary to uniquely specify the refrigerator and the computer reporting the results of each decision immediately.

Overall cycle parameters are strongly dependent upon these design decisions, which are themselves strongly interrelated. For example, the choice of stage temperatures and mass flow rates both have very non-linear effects on any cycle's compressor input power.

Since many of the design trade-offs cannot be recognized a priori, a fully automatic design procedure is, if not impossible, a difficult and costly approach. The interactive computer aided design proved to be particularly well suited for this type of problem. This approach shifts the mass of conceptually simple but tedious work to the computer leaving the engineer with the problem of recognizing and making trade-off decisions.

There are a number of other advantages in the interactive cycle design approach where the designer makes decisions and the computer provides rapid analysis based upon these decisions. Design decisions are made one at a time and can be changed as their effects become apparent. At each step of the process, the designer has the option of either proceeding with another design decision or revising his design decision and then proceeding. Finally, the designer can go back to an earlier decision point, revise it, and proceed with the new decision. Fig.2 is an example of the flow of the analysis. Note that the designer must respond to the interactive questions in order to proceed through the analysis of a refrigeration cycle. In the figure the user supplies inputs following a $ sign, and the computer program then provides outputs

3. COMPUTER PROCEDURE. The analysis procedure was implemented using a structured programming approach. The code consists of a main program and several layers of subroutines. The main program controls the cycle design process by executing the decisions made by the designer.

The highest level of subroutines consists of the basic building blocks of a refrigerator. Referring to Fig.2, building blocks of a typical cycle are Joule-Thomson block, expansion engine block, and compressor block. A building block subroutine contains the thermodynamic relations that describe the individual components of the block. For example, the Joule-Thomson block has two components: the Joule-Thomson valve and the load heat exchanger at 4.5K. In the analysis the Joule-Thomson valve is treated ideally as a constant enthaply device, i.e., fluid passing through a J-T valve neither gains nor losses energy. The 4.5K load is modeled as a device which absorbs energy at a specified pressure drop. In a similar fashion, the other building block subroutines are built up from the relationships that describes their components.

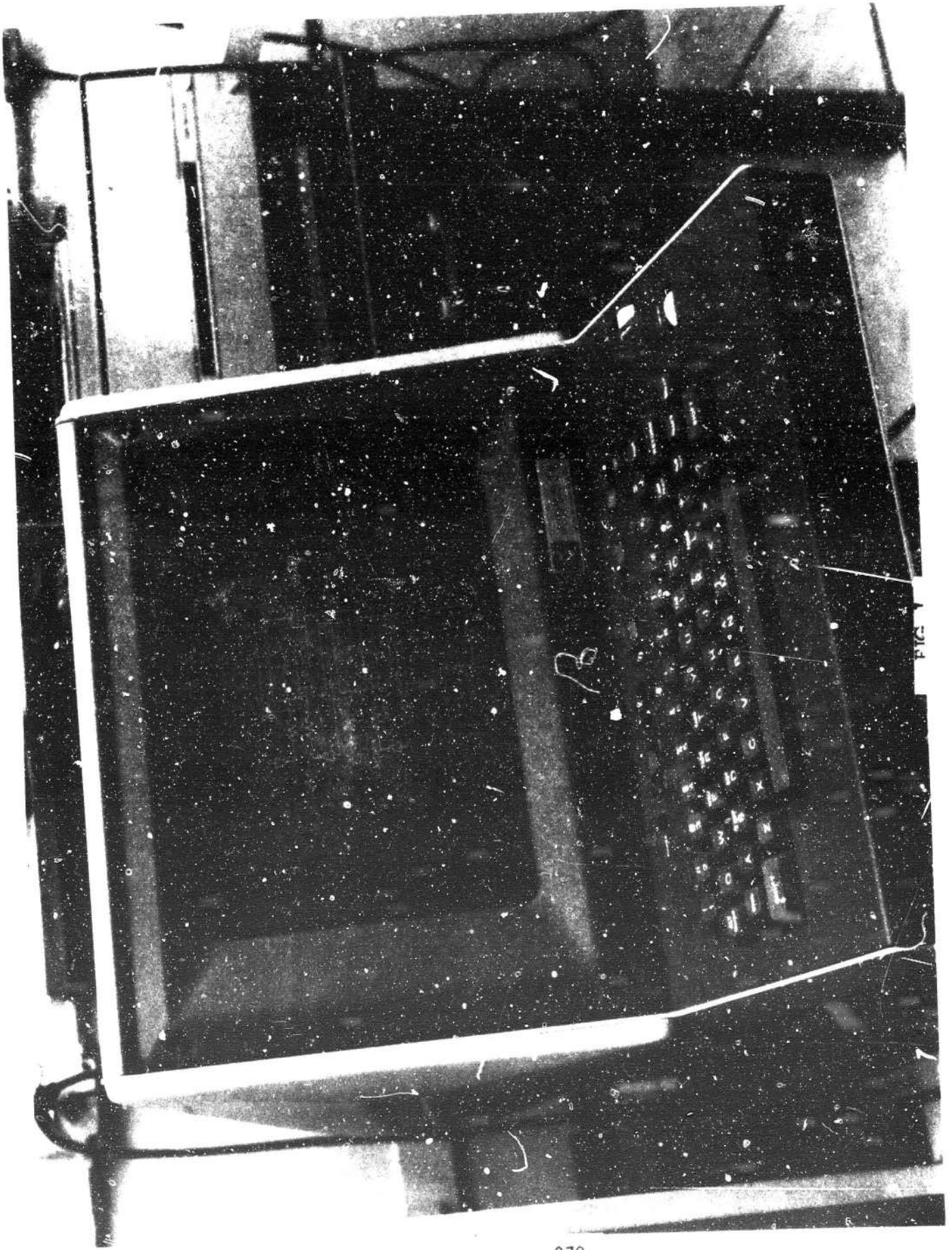
A list of the thermodynamic state points specifying the cycle is built up in memory. The building block subroutines. when called, add to this list, but they do not alter the previously completed parts of the design. This feature permits the designer to easily repeat a particular step until he is satisfied.

The building block subroutines determine thermodynamic state points by analyzing the relationships, adequate to characterize the block, from the terminal inputs and from the next level of subroutines. This level of subroutines consists of component characteristics and stored thermodynamic functions. The component characteristics subroutines are table look-up routines that return expansion engine efficiencies as a function of the inlet flow conditions. The stored thermodynamic functions subroutines consist of the functions given in Eq(1); these subroutines are also table look-up routines. Currently the thermodynamic properties of helium are used in the thermodynamic subroutines; but the algorithims of the subroutines are sufficiently general that the thermodynamic properties of any other refrigerants could be used instead of helium. The flexibility of the structured approach permits refrigeration cycles employing a refrigerant other than helium to be analyzed by changing only the thermodynamic data.

As an example of output, an optimized Claude cycle consisting of one expansion engine is shown graphically in Fig.4. A summary table of the state points is also available from an optimization run. This particular cycle was optimized by varying the cycle mass flow rates for constant heat exchanger effectiveness and compressor pressure ratio until the minimum power to the

compressor was obtained.

4. CONCLUSIONS. Optimized thermodynamic cycles for cryogenic refrigerators can be designed by a procedure based upon rapid computer analysis of design decision trade-offs. An interactive computer program has been written for this procedure, and it has proven to be useful for designing optimized helium refrigerators. The optimization procedure appears to be completely general. Refrigerator cycles employing refrigerants other than helium can be analyzed by replacing the thermodynamic properties of helium with those of the desired refrigerant. (Anyone interested in the details of the computer code may obtain a listing by writing to the authors.)



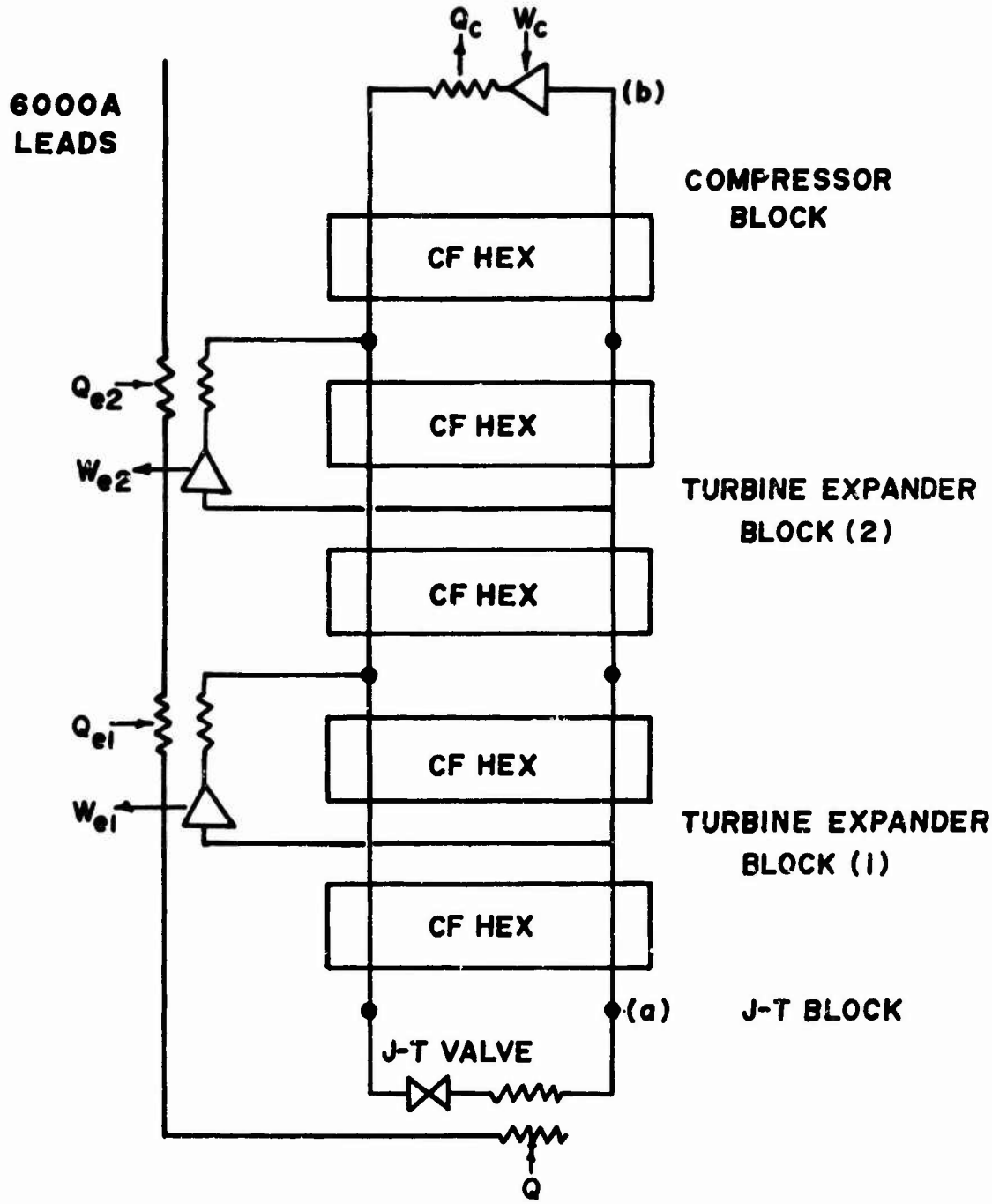

Q_c
W_c
(b)
6000A
LEADS
COMPRESSOR
BLOCK
CF HEX
Q_{e2}
CF HEX
W_{e2}
TURBINE EXPANDER
BLOCK (2)
CF HEX
Q_{e1}
CF HEX
W_{e1}
TURBINE EXPANDER
BLOCK (1)
CF HEX
(a)
J-T BLOCK
J-T VALVE
Q

FIG. 2

```
LOWER HEX T(HP-IN):
$ 18.
LOWER HEX EFFH=  .985 EFFT=  .811
NEW T  -1=NO +1=YES
$ -1
MASS FLOW IN TUR:
$ 29.
TUR EFF=72.984
POWER EXTRACT FROM TUR= 1663.65
TUR INLET T =  29.
HEAT LOAD AT TUR:
$598.
UPPER HEX EFFH=  .969 EFFT=  .971
UPPER HEX T(HP-IN)=  25.177
-1=BACK,0=REPEAT,+1=ADVANCE:
$  1
0=OUT,1=EX+UH+BH,2=EX+BH,3=BH+CMP:
$  3
CCCCCCCCCC          CCCCCCCCCC
HEX T(HP-IN):
$305.
HEX EFFH=  .984 EFFT=  .982
BACK TO EXP  -1=NO +1=YES
$ -1
COMP EFF EST:
$  .5
HEAT COMP(KW)= 140.0 MDOT(G/S)=  41.0 PR=  6.40
IDEAL ISOTHER AND ADB PWR(KW)=  48.8 AND  69.2
TOT CRYO HEX(M**3)= .0487 MASS(KG)=  131.
-1=BACK,0=REPEAT,+1=ADVANCE:
```

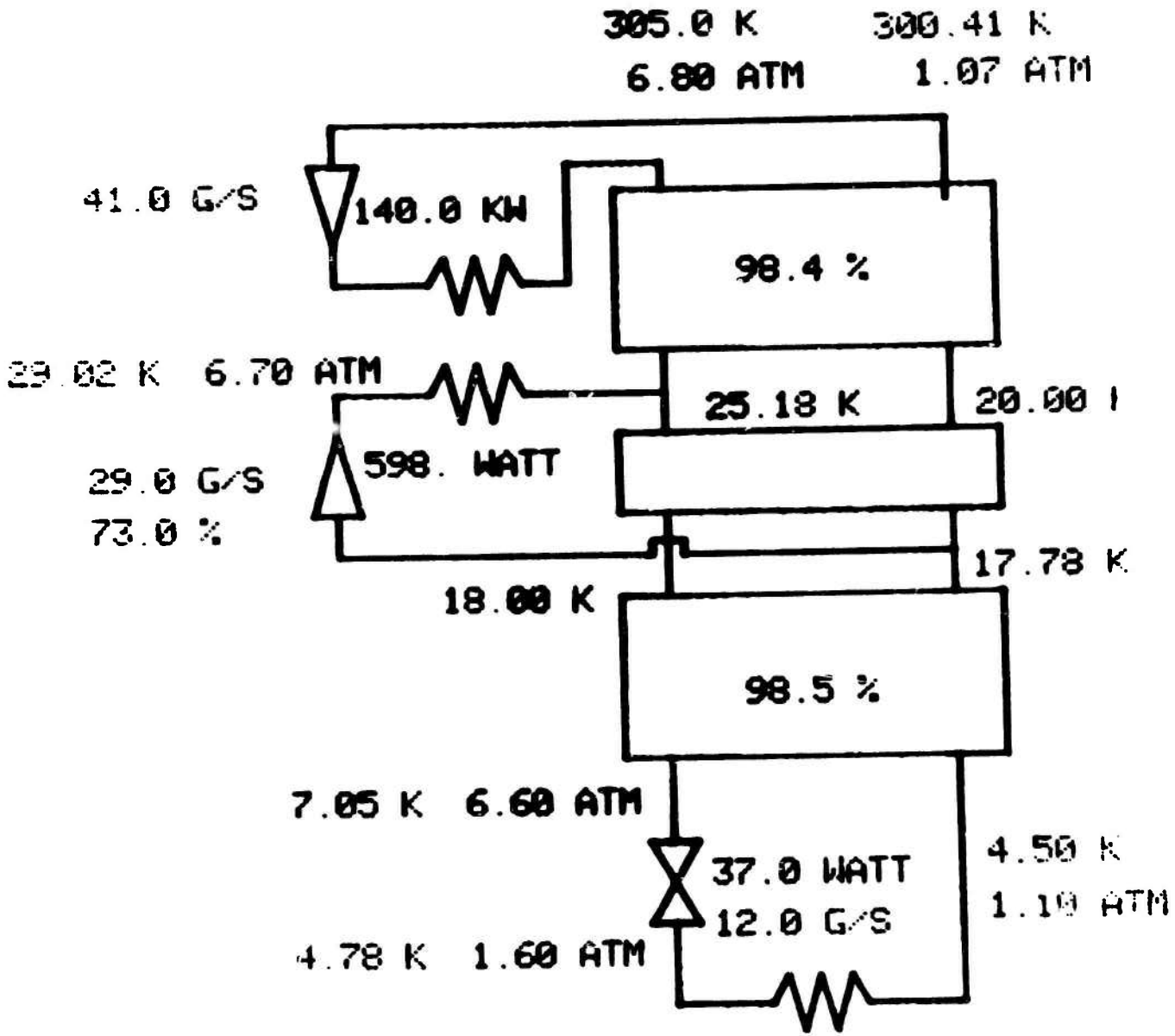

305.0 K
6.80 ATM
300.41 K
1.07 ATM
41.0 G/S
140.0 KW
98.4 %
29.02 K 6.70 ATM
25.18 K
20.00 K
598. WATT
29.0 G/S
73.0 %
17.78 K
18.00 K
98.5 %
7.05 K 6.60 ATM
37.0 WATT
12.0 G/S
4.50 K
1.10 ATM
4.78 K 1.60 ATM

FIG.4

COMPUTER AIDED X-RAY ANALYSIS OF SELECTED AMMUNITION MATERIALS

Fred Witt
Materials Engineering Division
Pitman-Dunn Laboratory
Frankford Arsenal
Philadelphia, PA 19137

ABSTRACT. The mathematics underlying two computer aided determinations of crystallographic anisotropy will be discussed, namely: (1) metal deformation texture, as depicted in computer-generated pole figures of copper shaped charge liners, and (2) the R-value index of metal drawability, as determined from automated pole figure analysis of cartridge brass. The armor penetrating ability of the shaped charge jet is considered to be critically dependent upon the annular symmetry of the metallurgical properties of the cone from which the jet is formed. The cartridge brass studies entail a comprehensive characterization of crystallographic texture to correlate formability with the anisotropy of strength of sheet materials. In both instances, the computer is used determine the location and concentration of those metallic grains which experience the maximum critical resolved shear stress under loading conditions. The predictive ability of the computer in assessing sheet metal formability will be demonstrated by comparing both x-ray and tensile test results for cartridge brass.

INTRODUCTION. Hard sphere models are a convenient means of illustrating the mechanism of plastic deformation in metals. An example is shown in Figure 1, where the atoms are represented as spheres stacked to form the face centered cubic (i.e., fcc) lattice. Copper, aluminum, and cartridge brass are some well known metals having this structure. The (111) plane is called the close packed plane since it contains the largest number of atoms per unit area. Similarly, the [110] direction is called the close packed direction because the packing density is greatest along that direction. From a microscopic point of view plastic deformation consists of a consecutive movement of the atom layers over one another by "sliding" in discrete crystallographic directions. This slip process proceeds from the movement of crystalline defects called dislocations. The Peierls (1) and Nabarro(2) equation predicts the shear stress (τ) required to move dislocations and is shown in Figure 2. It can be seen readily that τ is strongly affected by Poisson's ratio (ν) and the quantity D/S. Here D is the distance between adjacent slip

planes and S is the distance between atoms in the slip direction. Plastic flow proceeds most easily in those atomic planes and directions having a minimum value of τ. For a particular metal, Poisson's ratio is fixed; this means that slip occurs on those slip systems having maximum D and minimum S. In other words, slip occurs on {111} planes in <110> directions for fcc metals.

SCHMID'S LAW. The tensile behavior of crystals of different orientations can be compared by resolving the tensile stress into a component which lies along the slip direction in the slip plane. An examination of the equilateral hyperbola plot of Figure 3 reveals that the tensile yield stress varies greatly with the orientation of the slip plane and slip direction. Experience has shown that crystals plastically deform when the resolved shear stress reaches a critical value, τ_c, which is a constant for a particular metal. It is interesting to note that τ becomes zero when the tension axis is parallel or perpendicular to the slip plane. More important, the maximum shear stress is reached when the slip plane and slip direction are tilted 45° to the tension axis. For this case the value of $\cos\lambda \cos\phi$ is 0.5. Constant τ_c therefore represents a fundamental mechanical property of a metal and gives information on the mode of plastic deformation.

THE EQUIPMENT. To predict the behavior of metals under loading it was necessary to fabricate suitable x-ray counting and plotting equipment in order to determine the orientation of the slip planes and slip directions of the million or so grains which comprise typical test samples. This grain orientation data is usually plotted by means of the stereographic projection. Its development is shown in Figure 4.

In practice, the sample is rotated in the x-ray beam about the position labeled O. When Bragg's Law is satisfied the radiation comprising the diffracted x-ray beam OQ "flashes out" to the detector located on the reflection sphere at the longitude and latitude position defined by Q. From the direction of the vector OQ one can compute the orientation of the vector ON. The latter describes the pole of the atomic plane which produces the radiation registered by the detector. The colatitude and longitude positions of N are described by the angles β and δ respectively. The stereographic projection of N is first accomplished by connecting N with S. The intersection of the line NS with the equatorial plane is called the stereographic projection of the pole N, and is labeled P. By synchronizing the rotation speeds of the test specimen and the radiation detector it is possible to insure that the detector is sequentially positioned at the 5100 positions on the reflection sphere to receive the x-ray intensity diffracted from the test specimen. The X and Y coordinates of P are given by

$$X = 99 * \tan(\beta_n/2) * \cos(\delta_n/2)$$

$$Y = 59 * \tan(\beta_n/2) * \sin(\delta_n/2)$$

where β and δ are the angles defined previously. The numbers 99 and 59 are needed because the plotting symbols on the teletype are two thirds as wide as they are high. The recursion relationships for the 5100 values of β and δ are given by

$$\beta_n = \beta_{n-1} - k_1 * n * \Delta t \quad \text{and}$$

$$\delta_n = \delta_{n-1} + k_2 * n * \Delta t,$$

where k_1 and k_2 are determined by the rotating speed of the test specimen and Δt is the time between successive data points. The x-ray intensities recorded at the 5100 positions of Q are placed in a two dimensional array whose indices are determined by the X and Y values of P. The array and its indices are defined as

$$JC = X + 101$$

$$JD = 61 - Y$$

Array (JD, JC) = x-ray intensity at Q.

The instrumentation to rapidly acquire, analyze, and plot the x-ray information describing the orientation relationships among the grains which comprise the test specimen is shown in Figure 5. As the coupon is rotated in the x-ray beam for 85 minutes, the intensity readings are taken "on-the-fly" by the output interface which can take and store the measured values of the diffracted intensity. The dead time of this module is 10^{-9} seconds per data point. The intensity values are converted to the required output code and fed in bit-serial-form, at 300 baud, to a "cassette" magnetic tape unit that records the data at 800 bpi on a standard 70,000 character Philips tape. The high speed teletype provides a printed copy of the x-ray data as it is recorded on the magnetic tape unit. Both are capable of accessing a remote digital computer through the model.

THE COMPUTER PROGRAM. The Fortran SOURCE program which is used to analyze the data and plot the finished pole figure is an improved version of one kindly provided by Dr. Glen A. Stone of the South Dakota School of Mines and Technology. The program requires an octal field length of 153600 and runs in about 30 seconds on a CDC 6500. The finished output consists of three pole figure plots. Each plot contains

the same information but is presented in a different form. Plot #1 contains the reduced input data without any attempt at smoothing. Plot #2 treats the input data of plot #1 by filling the matrix and smoothing the data. In making plot #2, the computer program allows the preselection of up to twenty specific iso-random values with selected tolerances. Plot #3 is probably the most useful one for drawing highly detailed iso-random level contour lines. It allows any one of 64 single alphanumeric symbols to be placed in each position of the 120 by 200 plot matrix, in 64 half-random level steps.

An example of these three plots for a shaped charge ammunition component is shown in Figures 6-8, respectively. The 'spiral' pattern of Figure 6 corresponds to simultaneous rotation of β and δ. The positions of the R's on the plot are true positions of the recorded x-ray intensity values. Figure 7 illustrates the case where plots of the 0.5, 1.0, 2.0, 3.0, and 4.0 random levels are desired. Turning now to Figure 8, the interface between X and 1 is exactly the 0.5 random level of Figure 7; between 1 and / exactly 1.0 random; between / and 2 exactly 1.5 random; and between 2 and / exactly 2.0 random, etc.

Representative {111} and {220} stereographically projected pole figures for a shaped charge liner are shown in Figures 9 and 10. The plotted x-ray intensity readings specify the ratio of the volume of material in the liner to the volume of material with the same orientation in a randomly oriented sample.

While {220} pole figures provide information on the "directions" available for plastic deformation, it is actually the {111} planes which undergo the plastic flow. The circle shown in Figure 9 represents the locus of points describing the {111} planes tilted 45 degrees to the surface of the liner's outside wall. As the detonation waves move down the surface of such a liner, it is reasonable to expect that those {111} planes tilted 45 degrees to the liner's surface feel the effect of the pressure and proceed to move in < 110 > directions. If the {111} planes are in non-optimum orientations, then the overall effectiveness of a preferred collapse direction in the liner wall is reduced.

FORMABILITY STUDIES. The press formability of a metal is described in terms of its drawability and stretchability. Drawability is related to crystallographic texture and is a measure of the metal's resistance to thinning in the short transverse direction. It is described by the plastic strain ratio R, defined as $R = \epsilon_w/\epsilon_t$, where ϵ_w and ϵ_t are true width and thickness strains respectively. For most metals, R varies with test direction in the plane of the sheet. Stretchability, however, describes the ability of a metal to be stretched, under biaxial tension, to conform to the contours of the punch.

A good deep drawing material exhibits a high resistance to thinning coinciding with easy plastic flow in the plane of the sheet. The differences in width and thickness strains observed during plastic flow are related to the orientations of the slip directions and slip planes of the grains which comprise the sheet material. Since unit slip along any of the slip directions can be resolved into components of width strain and thickness strain one can calculate the plastic strain ratio from an examination of the (220) quantitative pole figure, which describes the position of all $<110>$ slip directions. If a tensile specimen from sheet copper is loaded in tension along its longitudinal axis, the directions of the maximum resolved shear stress generate a cone whose axis lies along the tension axis and whose semi-apex angle is 45 degrees. For small plastic deformations, the operative crystallographic planes are those lying closest to the cone of maximum shear stress. With increasing strain, deformation occurs in those grains less favorably oriented. The intersection of the cone of maximum shear stress is shown in Figures 11 and 12a, for the cases where the tension axis is oriented 0°, 45°, and 90° to the rolling direction of sheet material. In Figure 12b, the shaped charge liner case is shown where the intersection of the cone of maximum shear stress describes a circle centered on the pole figure.

To compute the plastic strain ratio R from the pole figure data the Fortran Source program described earlier is expanded so that the 940 x-ray intensities of the $\{220\}$ reflection lying within a $\pm 10^{\circ}$ band of the cone of maximum shear stress are weighted in accordance with the formulas given in Table I. The formulas listed above the pole figures describe the circular arc produced by the intersecting cone of maximum shear stress. Beta and delta are the colatitude and longitude of the $\{220\}$ poles respectively.

Table I - Weighting Factors for Computing Average Anisotropy Contributions From Different Crystallographic Orientations

Angle Between Tension Axis and Rolling Direction	Weighting Factor
0°	$\tan \omega = \tan \beta \cos \delta$
45°	$\tan \omega = \frac{1}{\sqrt{2}} \tan \beta \ (\sin \delta - \cos \delta)$
90°	$\tan \omega = \tan \beta \sin \delta$
Shaped charge liner case	$\tan \omega = \cot \delta$

It can be seen that the absolute value of the weighting factors range from zero to infinity. An inspection of the pole figure plots reveals that the weighting factor is zero when the component of unit slip lies totally in the short transverse direction; it approaches infinity when slip occurs entirely in the plane of the sheet material. As a test of the validity of this approach, in-situ determinations of R obtained from standard tensile tests on cartridge brass were compared with R predicted from the quantitative {220} pole figures. The results, taken from reference (4), do establish the validity of this approach and are shown in Table II.

Table II - A Comparison of $\overline{R}^*$ Values from Tension Tests and Values Derived from Pole Figures

		Pole Figure Results	
Alloy	Tension Test	$R = \tan \omega$ Criterion	$R = \tan^2 \omega$ Criterion
G	.93	.94	.94
H	1.10	.90	.90
0-1	.90	.93	.94
P-1	1.06	.97	.99
0-2	.97	.97	1.00
P-2	1.12	1.00	1.03
U-2	.84	1.02	1.07
V-2	.92	1.02	1.08

$$\overline{R}^* = \frac{R_0 + 2R_{45} + R_{90}}{4}$$

The column titled $R = \tan \omega$ criterion contains $\overline{R}$ predictions for the case where the width and thickness strains are assumed to depend only on the slip directions and not on the orientation of the active slip plane. The column $R = \tan^2 \omega$ criterion treats the more general case, i.e., where plastic flow occurs on the {111} planes in $< 110 >$ directions.

To provide an index of the textural asymmetry for the shaped charge liner case, the {220} pole figure in Figure 11b is first divided in half by a vertical line. Two R values are computed; one for the left side of the pole figure (R_l) and another for the right side (R_r). By letting $\Delta R = R_l - R_r$, it is readily seen that ΔR approaches zero for symmetrical pole figures. ΔR can take on positive or negative values depending on the sense of the textural asymmetry of the grains comprising the liner.

SUMMARY. Evidence is presented for the existence of preferred slip planes and slip directions in fcc metals.

Equipment is described which affords an approach to shaped charge liner improvement through texture control. This approach recognizes those textural components useful for providing efficient and continuous jets.

Quantitative pole figures may be used to describe preferred orientation in deep drawn ammunition components and shaped charge liners.

Metal formability may be assessed with precision and speed by computing the plastic strain ratio from the pole figure data by averaging the anisotropic contributions from grains of various crystallographic orientations.

The effects of changes in mechanical and thermal processing on the R value may be used as a tool to assess material drawability, both for research and production quality control purposes.

REFERENCES.

1. Peierls, R., Proc. Phys. Soc. 52, 34, (1940).

2. Nabarro, F. R. N., Proc. Phys. Soc. 59, 256, (1947).

3. Schmid, E. and Boas, W., Plasticity of Crystals, F.A. Hughes and Co., 1950.

4. Witt, F. and Lawley, A., - "The Determination of Normal Anisotropy from Pole Figures," Final Report INCRA Project No. 129 - Drexel University (1973).

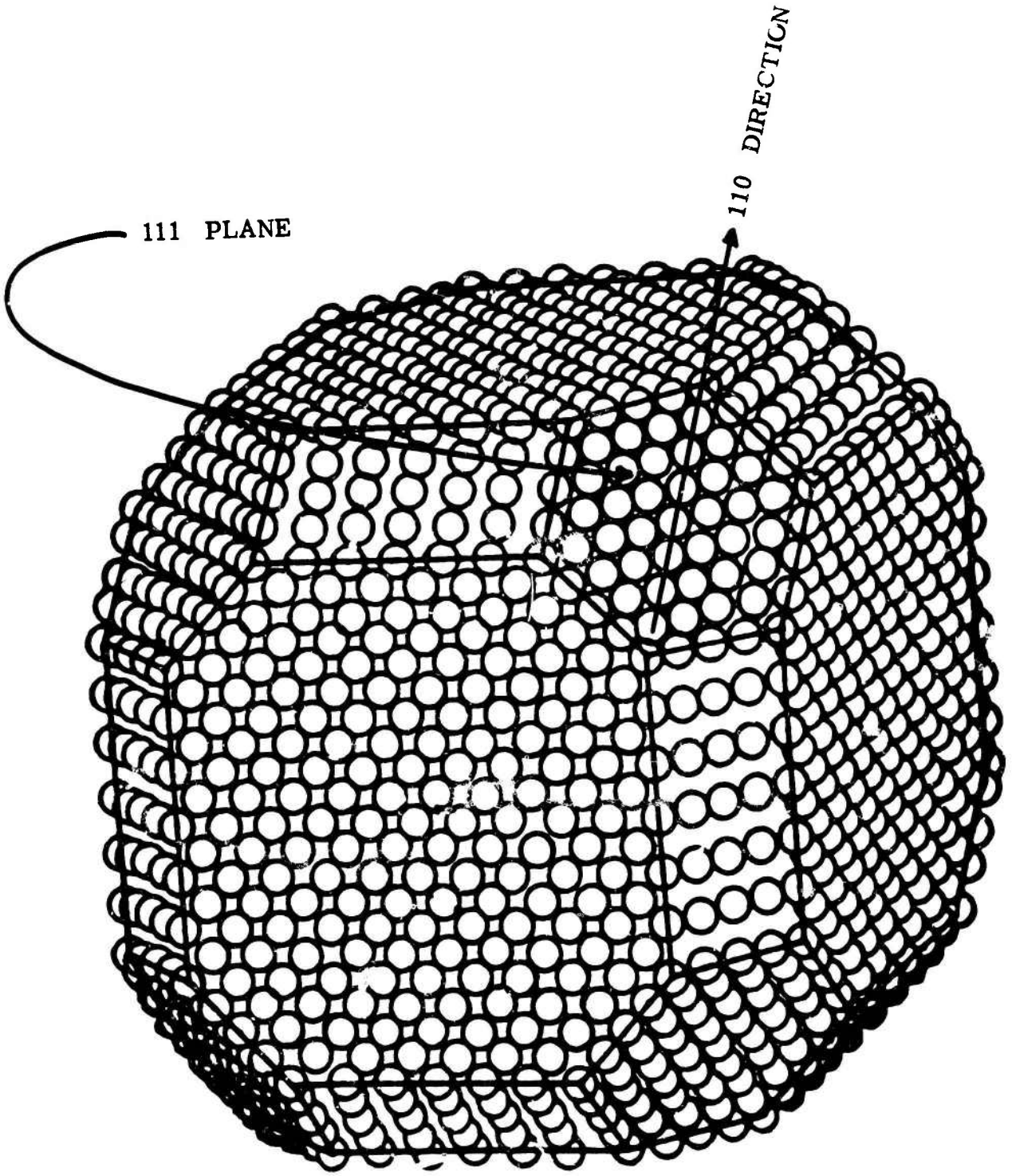


Figure 1. Hard sphere model of an FCC metal.

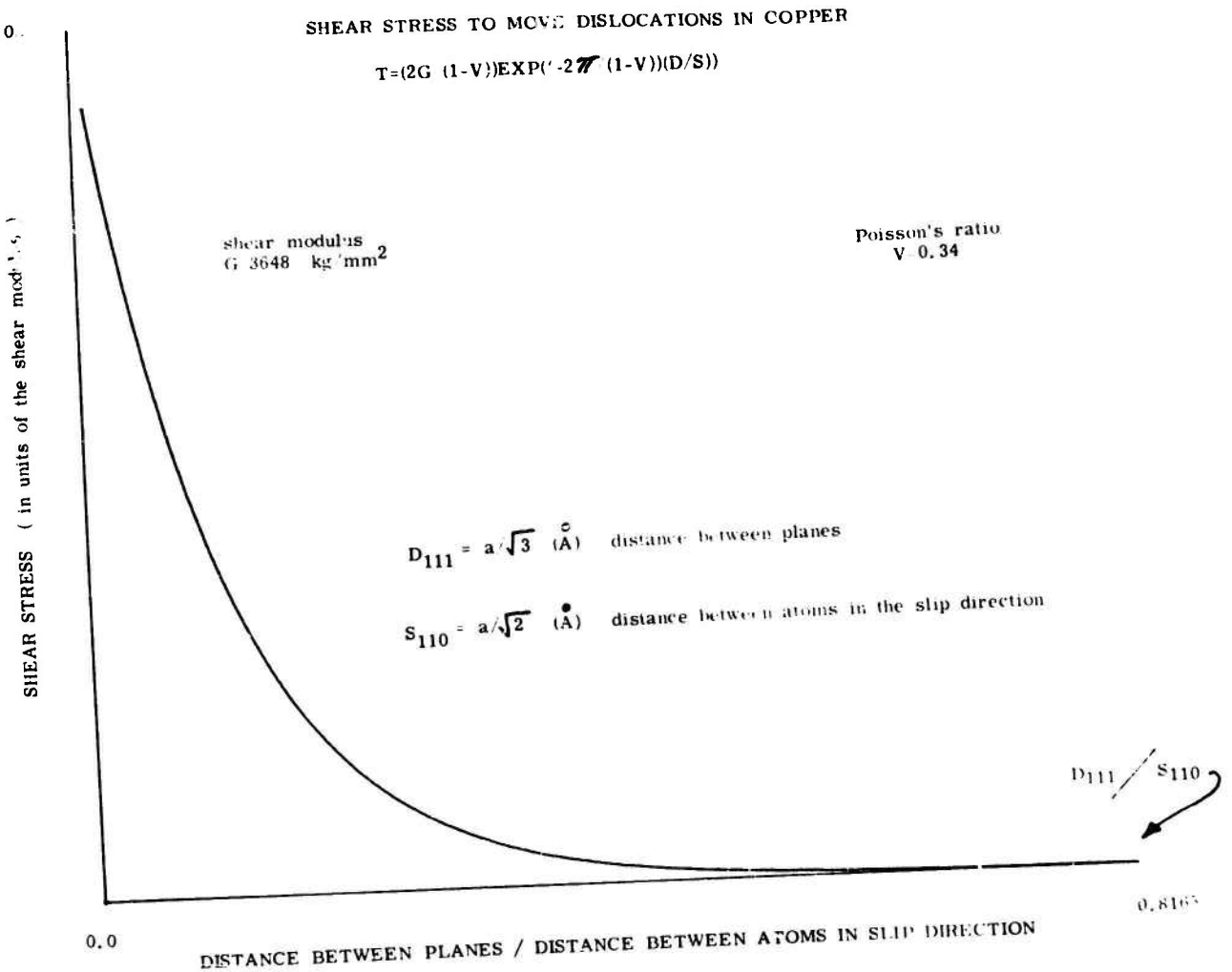


Figure 2. Shear stress to move dislocations in copper.

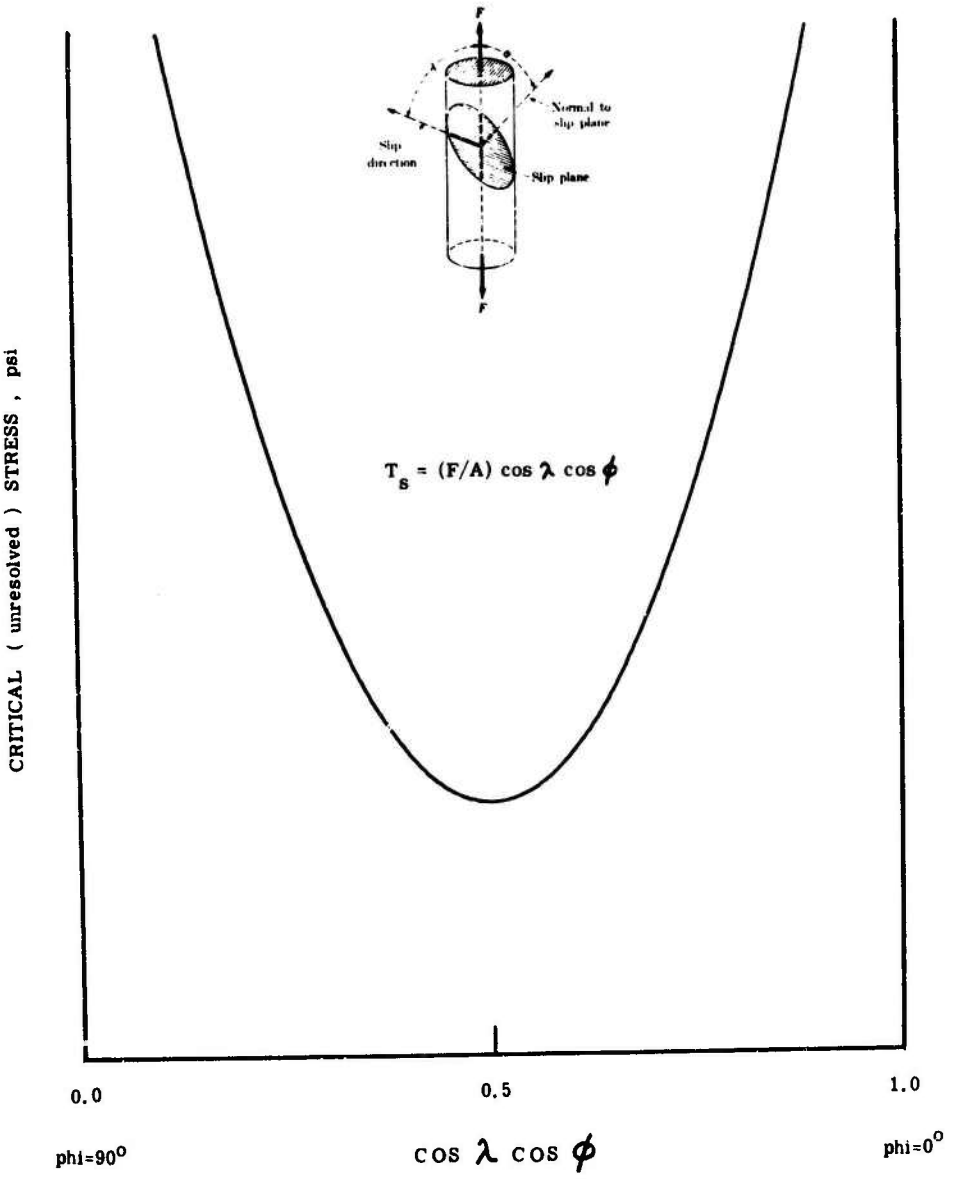


Figure 3. Variation of critical unresolved tensile stress with cos λ cos φ .

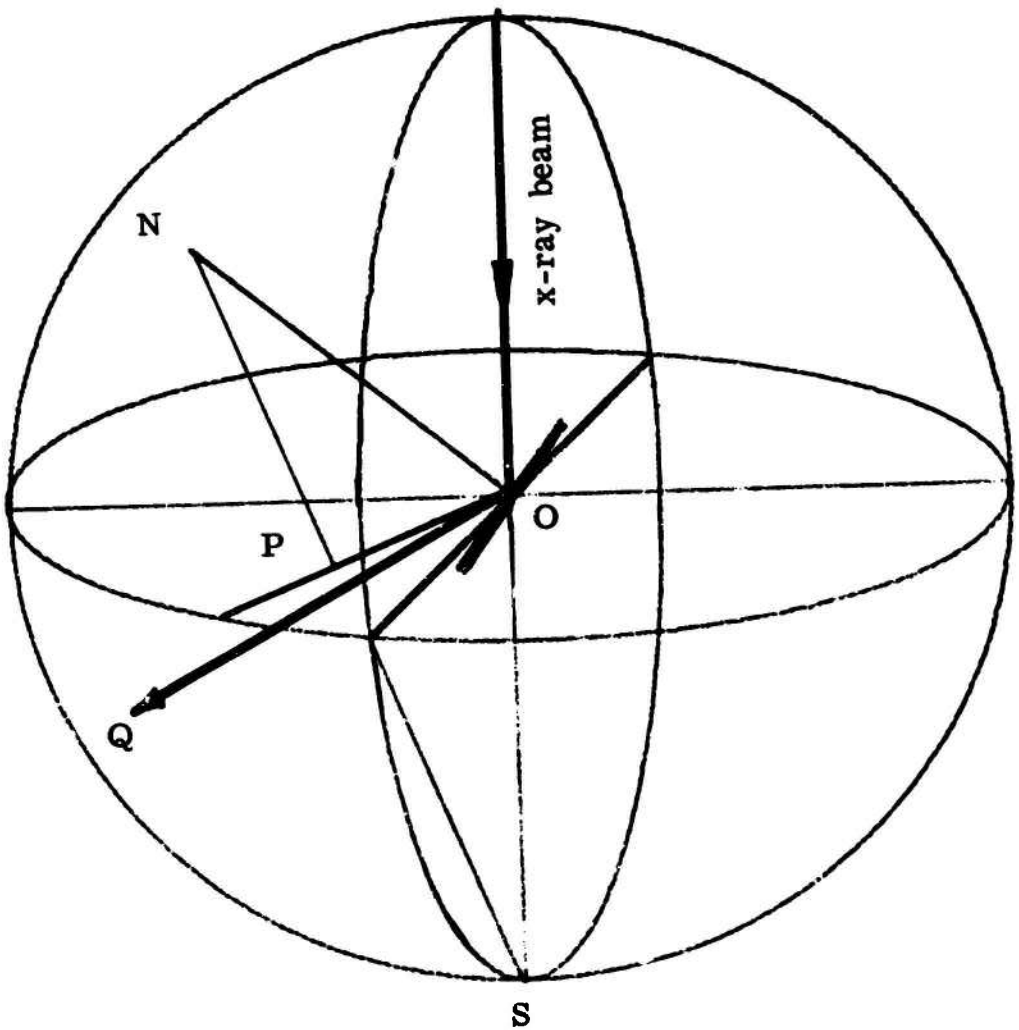


Figure 4. The relation between the diffracted beam OQ and the stereographic projection P of the normal ON to the reflecting plane.

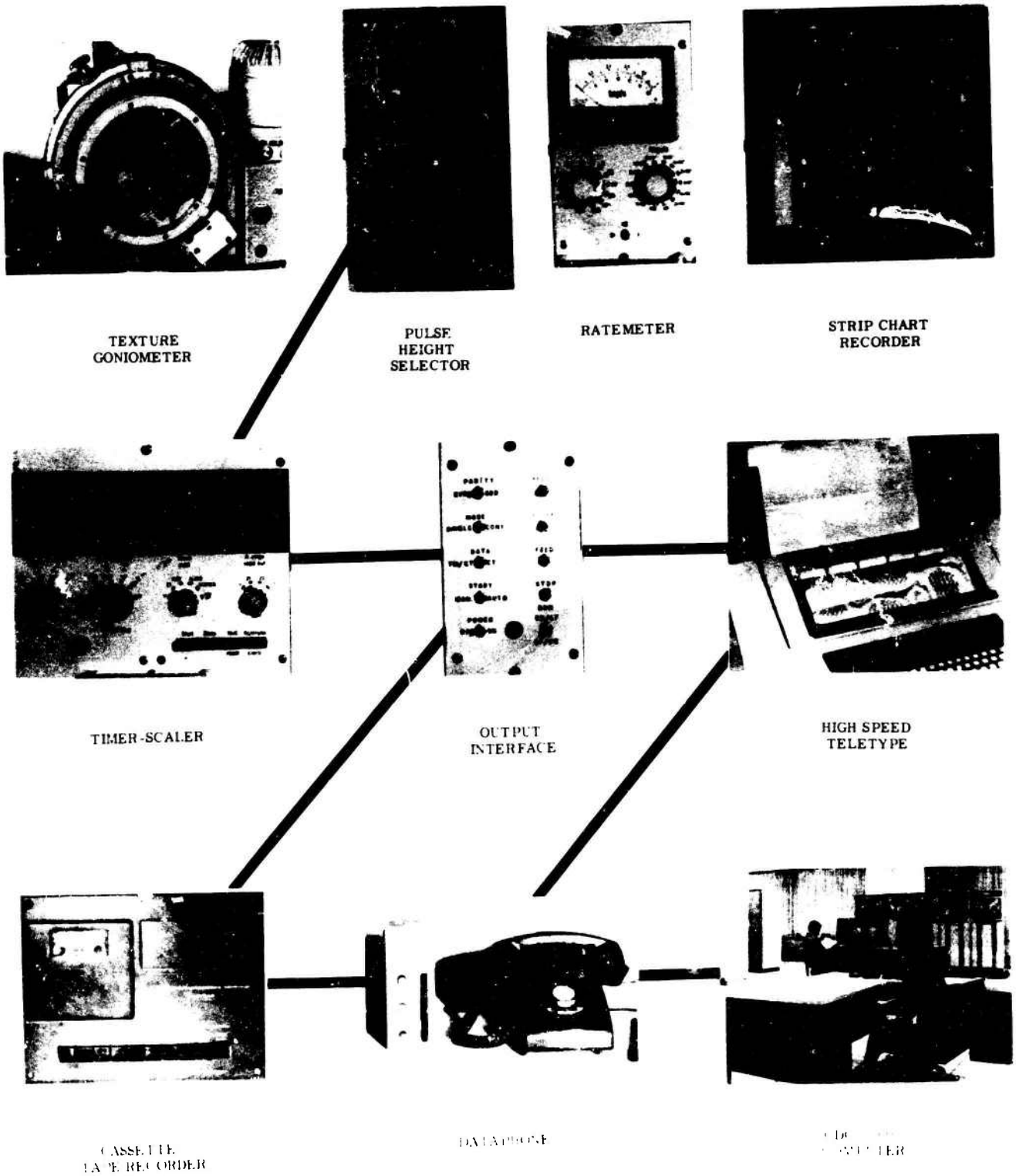


Figure 5. Block diagram of major components for automated texture diagram determination.

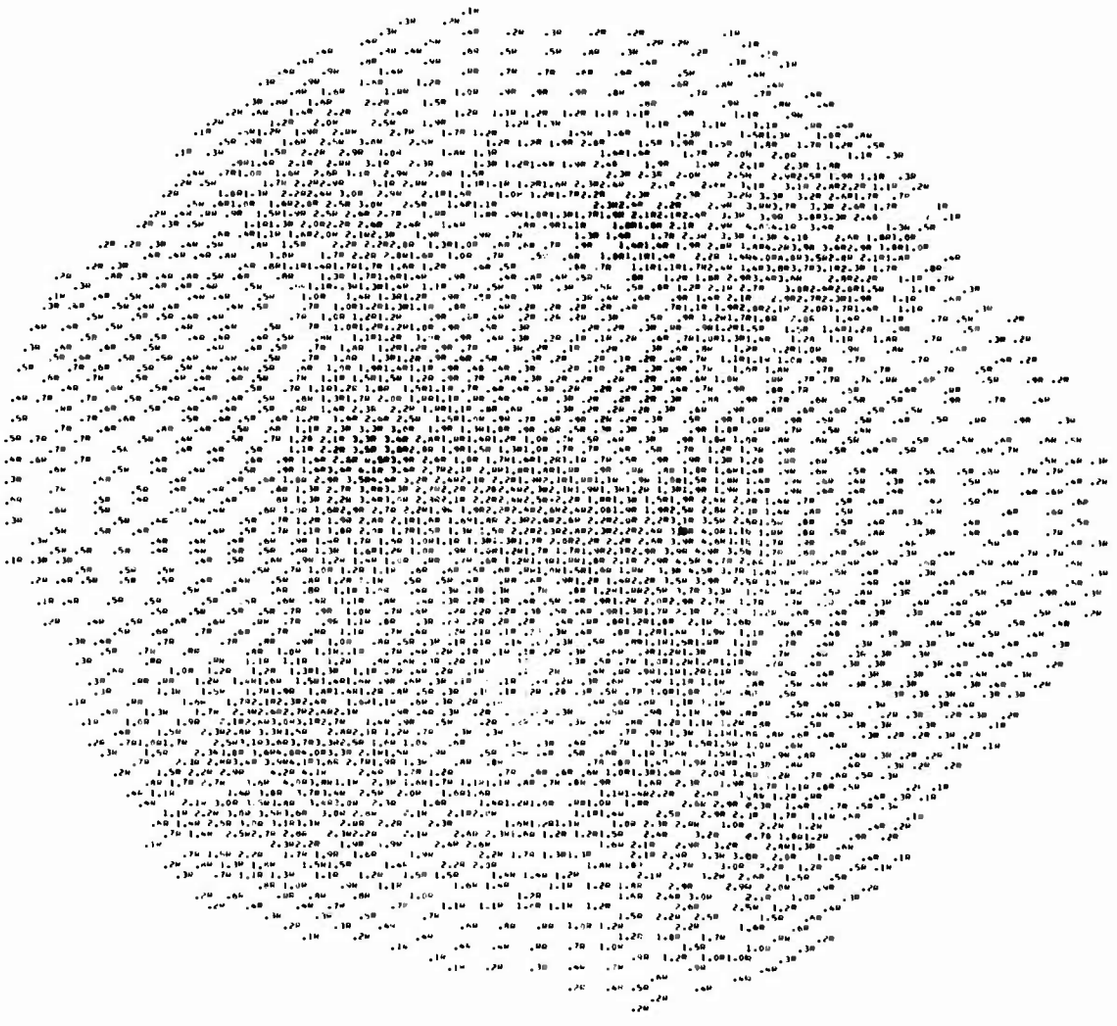


Figure 6. (220) pole figure illustrating computer output for plot # 1.

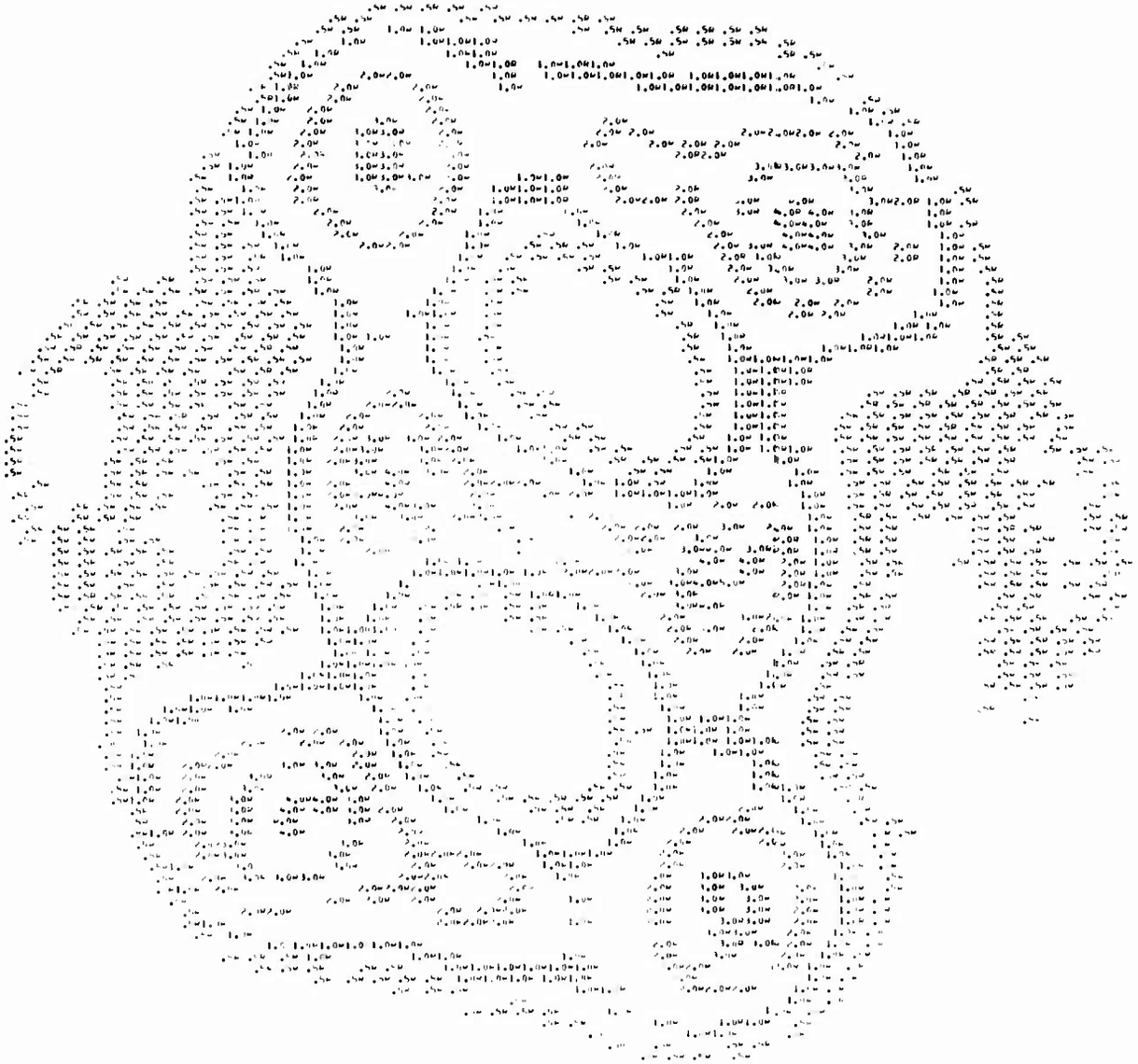


Figure 7. (220) pole figure illustrating computer output for plot # 2.

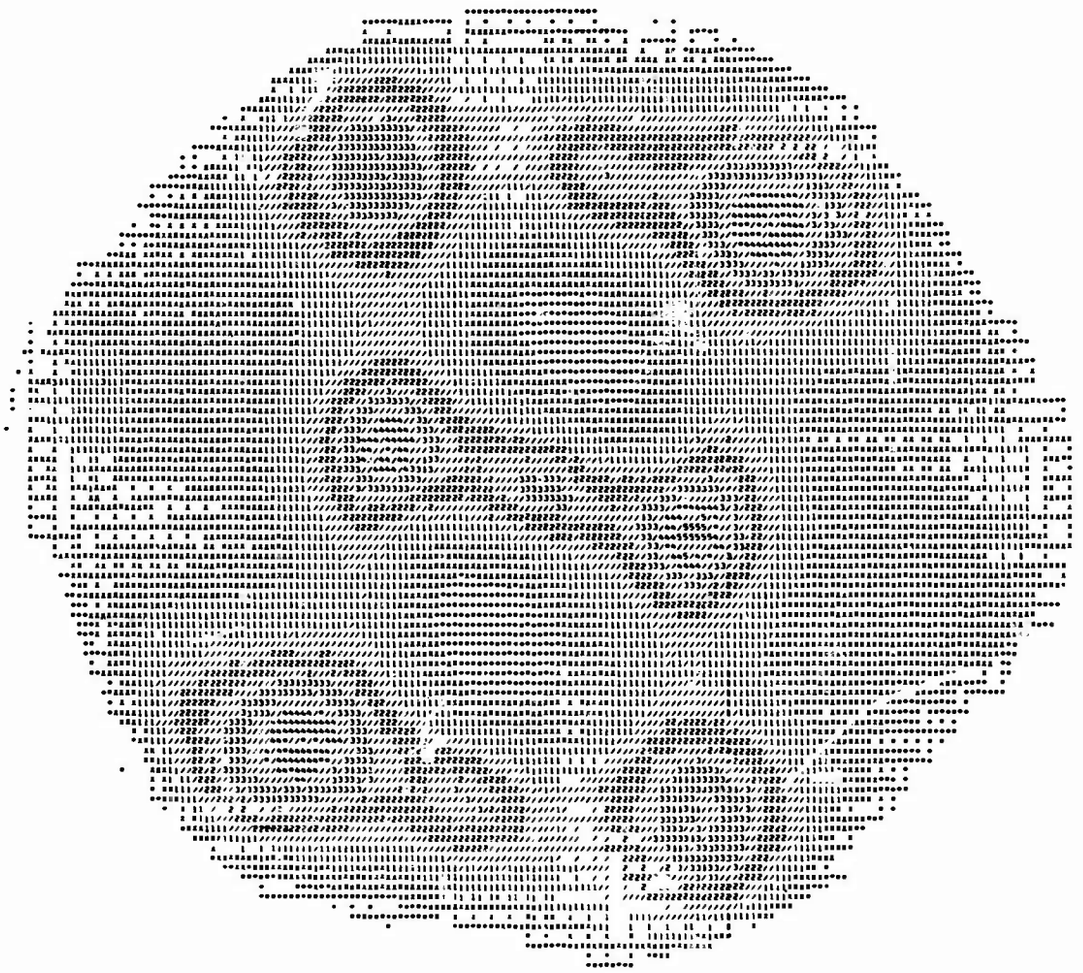


Figure 8. (220) pole figure illustrating computer output for plot # 3.

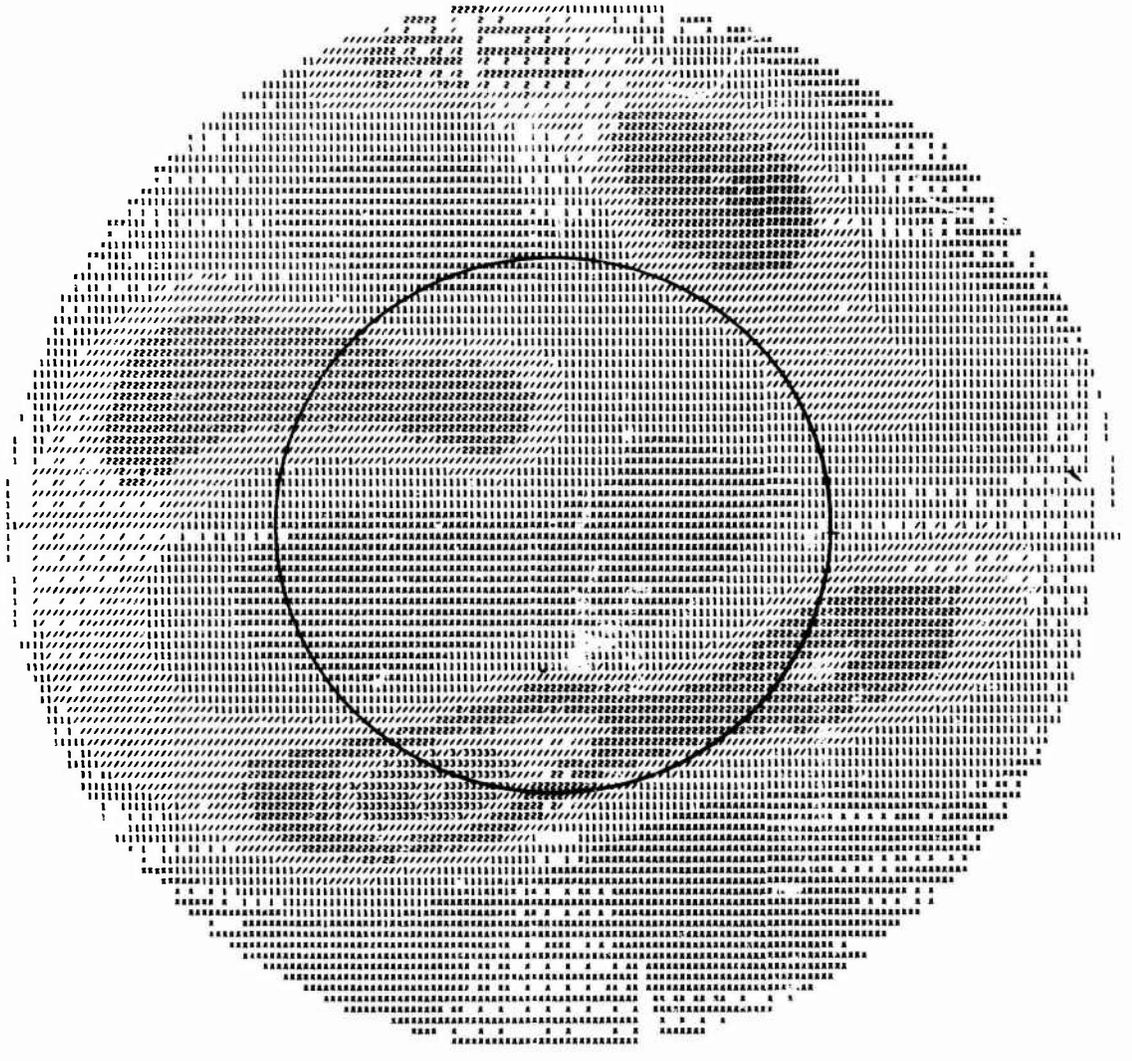
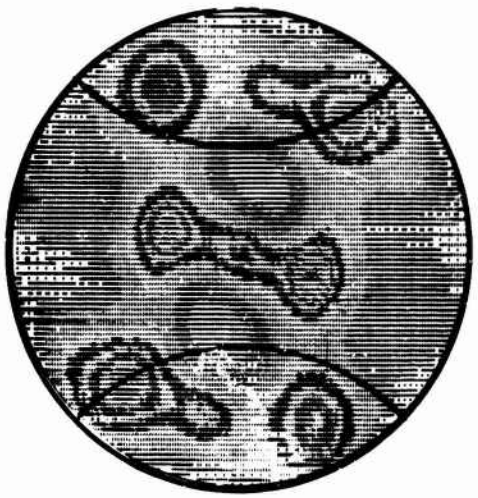


Figure 9. (111) pole figure for a shear spun copper liner.

Figure 10. (220) pole figure for a shear spun copper liner.

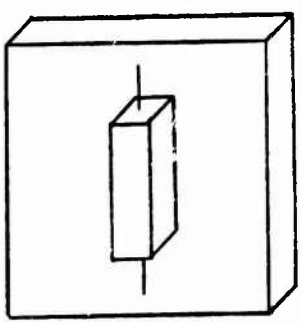
$\sin\beta \sin\delta = \pm \cos 45^{o}$

$\sin\beta \sin\delta + \sin\beta \cos\delta = \pm\sqrt{2} \cos 45^{o}$



$\tan w = \tan\beta \cos\delta$

$\tan w = \frac{1}{\sqrt{2}} \tan\beta (\sin\delta - \cos\delta)$



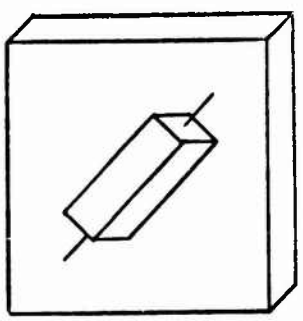


Figure 11. Intersection of the cone of maxium shear stress for 0 and 45 degree case.

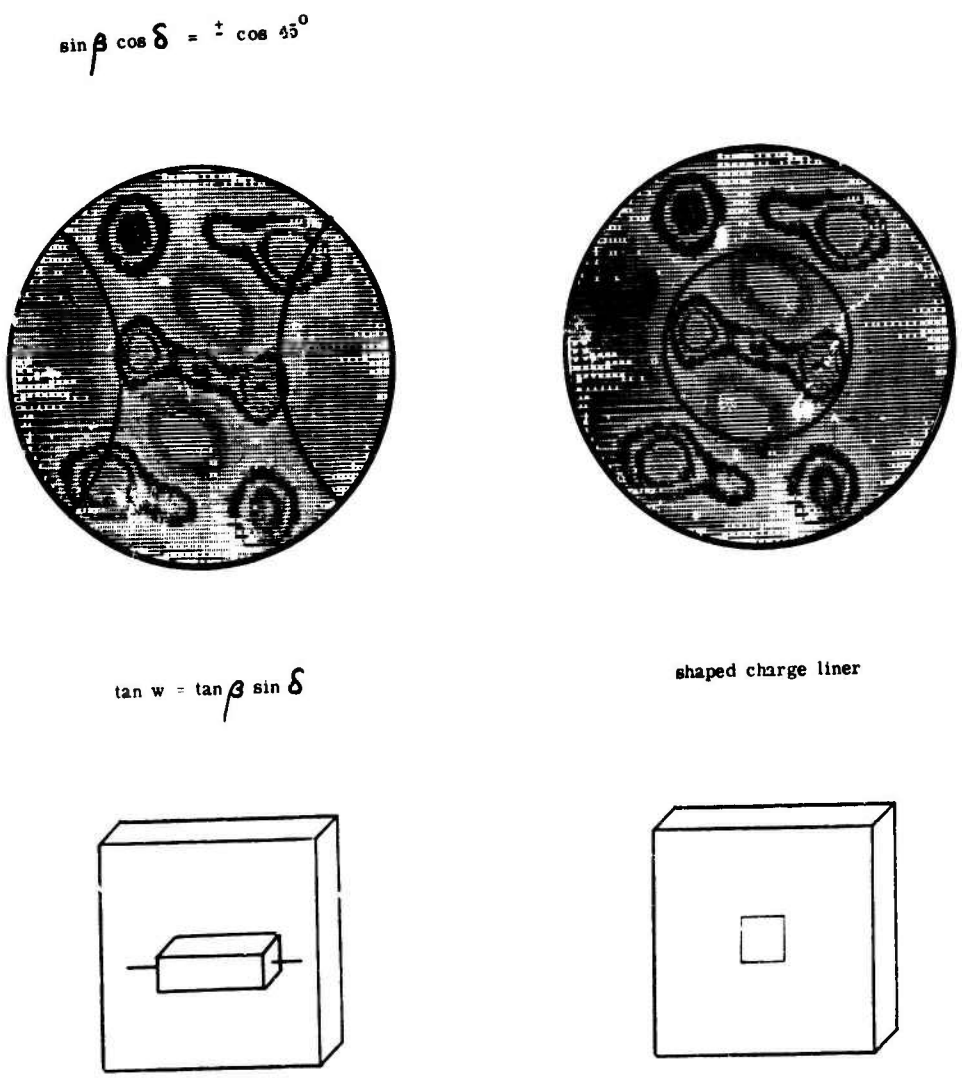


Figure 12. Intersection of the cone of maxium shear stress for 90 degree and shaped charge liner case.

A BACKWARD SOLUTION COMPUTING MISS DISTANCE FROM INPUT ERRORS TO GUN AIR DEFENSE SYSTEMS

T. H. Slook
Fire Control Development and Engineering Directorate
Frankford Arsenal, Philadelphia, Pennsylvania

INTRODUCTION

The operation of the conventional gun air defense system can be described with reference to Figure - I. In the conventional system, the gunner tracks the target with some form of sighting system. This tracking system provides some or all of the following target data as a function of time:

D_o - present position slant range

$\dot{D}_o$ - present position slant range rate

A_o - present position azimuth

$\dot{A}_o$ - present position azimuth rate

E_o - present position elevation, and

$\dot{E}_o$ - present position elevation rate

The tracking data collected are fed to some form of computer where they are processed to account for target velocity and the exterior ballistics of the projectile. This computer can be as simple as a man's gross estimate of the target motion and the required ballistics or it can be a full fledged digital solution, employing a myriad of sensor inputs. In any case, by some means, sophisticated or unsophisticated, a predicated weapon line is established and by means of a movable gun turret the projectile is fired along this established line.

Figure I gives a pictorial representation of the anti-aircraft problem of level flight and constant speed. It also explains the symbols used in the mathematical discussion found in the sequel.

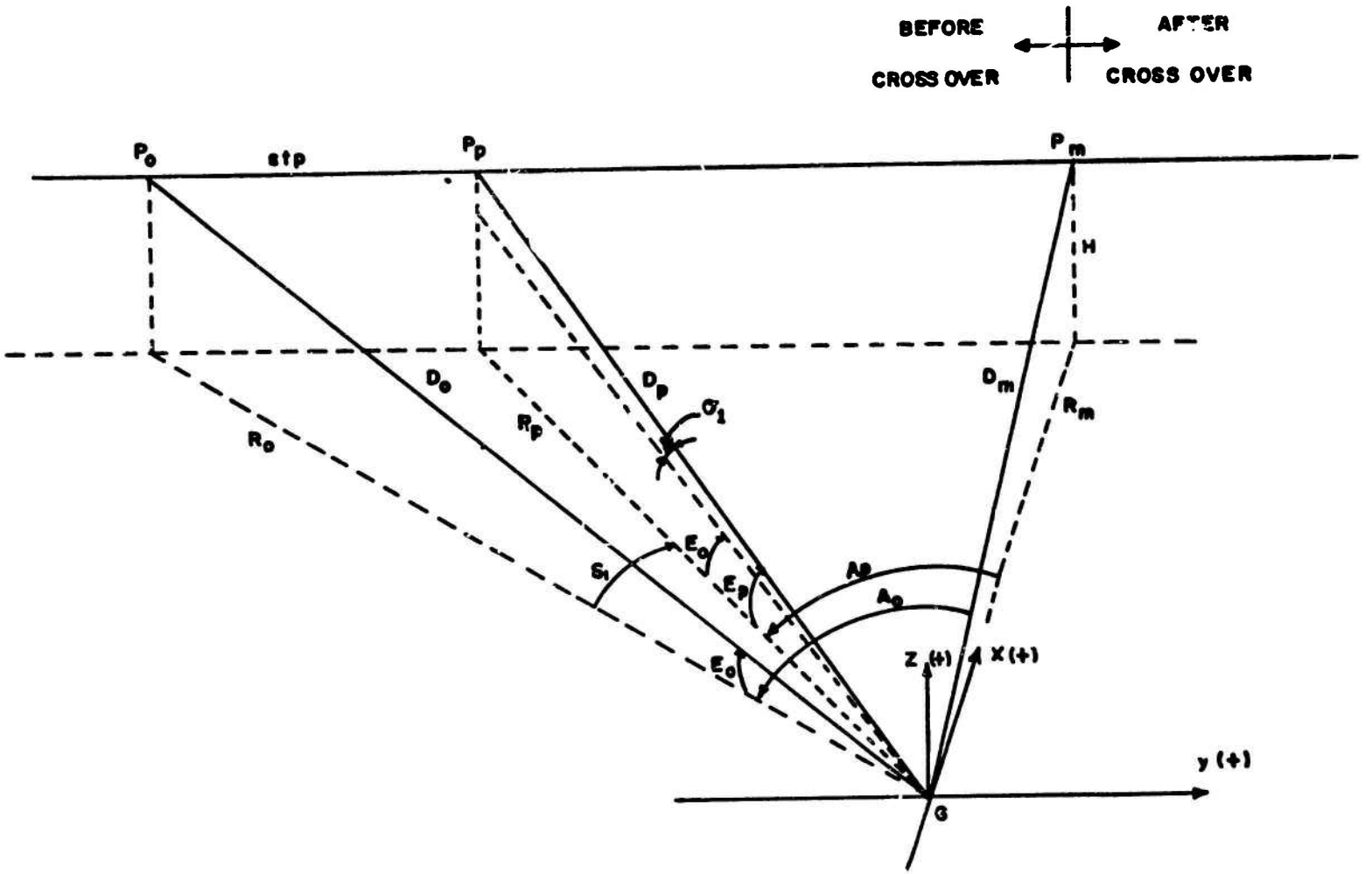


Figure 1. 3-D Fire Control Prediction Geometry of Level Flight Constant Velocity Targets

Let the number triple (s, H, Dm), where

s - ground speed of aircraft,

H - height of aircraft above gun plane, and

Dm - cross-over slant range,

describe a level flight, constant speed target path with respect to the gun which is the origin of the coordinate system. Observe that to hit a moving target from a fixed gun position, one must lead the target. Thus, the gun is positioned in azimuth and elevation so that the projectile impacts with the target at the predicted point Pp. To find the predicted point, one must determine:

1. the speed of the target

2. the path of the target, and

3. the trajectory of the projectile during the time interval from point Po to point Pp.

The tracking data is used to estimate 1. and 2. and BRL supplies the projectile trajectories as ballistic tables, ballistic differential equations, or ballistic algorithms. The computer operates on the tracking data and ballistic data by solving a non linear system of equations and generates, in real time, information which positions the gun (or the sight) so that the projectile should impact the target at Pp.

Many of the projectiles will miss the target by some amount. This miss will occur because of errors in the tracking data, errors in the projectile ballistic fit, prediction algorithm in the computer, etc. Thus, one measure of effectiveness for gun air defense systems is projectile - target miss distance MD resulting from error in the system. In our case, the MD

results only from errors in tracking data. For the present, it will suffice to think of MD as the minimum distance between projectile and target.

Instead of inputing tracking data, ballistics and errors in the inputs then solving a system of non linear equations for MD, we compute MD by inputing Dp, Ap, Ep, a ballistic fit and errors in tracking data. This computation avoids the long and tedious solution of the system of non linear equations. This backward solution is presented in the following section.

A FORTRAN program is presented listing tables and discrete graphs of MD for equal intervals of T_o, D_o, and D_p as functions of

ΔD_o - error in present position slant range,
$\Delta \dot{D}_o$ - error in present position slant range rate,
ΔA_o - error in present position azimuth,
$\Delta \dot{A}_o$ - error in present position azimuth rate,
ΔE_o - error in present position elevation,
$\Delta \dot{E}_o$ - error in present position elevation rate,
C_g - cant of gun trunnion, and
T_r - weapon bore tilt,

for different ballistics trajectories. Since ballistic effects change the time of flight of the projectile, a second FORTRAN program is given relating MD to Δt_p, the error in time of flight, for equal intervals of T_o, D_o and D_p.

The above miss distance information exhibits the optimum errors in the outputs of an antiaircraft fire control system for expected errors in the inputs. In other words, these programs provide information which is of value in measuring the effectiveness of a fire control system.

MISS DISTANCE
AS A FUNCTION OF PRESENT POSITION DATA ERRORS

Let (s, D_m, H) describe a target with respect to the gun as center of the coordinate system (see Figure 1).

The FORTRAN program - Miss Distance as a Function of Unit Input Errors - does not accept the ballistic data in tabular form. The data must be entered in function form; i.e., $t_p = f(D_p, E_p)$. The "fan" fit to ballistic data (described in Appendix B) generates t_p as a function of D_p and E_p, and has the form

$$t_p = a_1 + a_2 D_p + a_3 D_p^2 + a_4 E_p + a_5 E_p D_p + a_6 E_p D_p^2 \qquad (1)$$

where a_k ($k = 1, 2, 3, 4, 5, 6$), depends on the projectile used and the units of measure for t_p, D_p, E_p. When the given standard ballistic data indicates that t_p is a function of D_p (E_p has little effect on t_p), a least square polynomial fit has the form

$$t_p = \sum_{k=1}^{m+1} a_k \, D_p^{k-1} \qquad (2)$$

where the a_k again depends on the projectile and the units of measure used. In the FORTRAN example of the sequel, the 35 mm Oerlikon projectile was fit by the latter method, using s polynomial of degree five. For a given partition of the D_p interval and the corresponding value of E_p, determined by the equation

$$\sin E_p = \frac{H}{D_p} \qquad (3)$$

one computes the variable t_p using Equation 1 or 2.

Time (T), measured in seconds, is assumed negative before cross-over and positive after cross-over. Times T_p and T_o, corresponding to predicted position point P_p and present position point P_o, respectively, are given by equations:

$$T_p = \pm \frac{\sqrt{D_p^2 - D_m^2}}{s} \tag{4}$$

$$T_o = T_p - t_p \quad . \tag{5}$$

For the given target path and each selected predicted position point (D_p, A_p, E_p), where A_p was computed from

$$\cos A_p = \pm \frac{\sqrt{D_m^2 - H^2}}{D_p \cos E_p} \tag{6}$$

the following present position data D_o, $\dot{D}_o$, A_o, $\dot{A}_o$, E_o, $\dot{E}_o$ were computed using equations

$$D_o = \sqrt{D_m^2 + (s\,T_o)^2} \tag{7}$$

$$R_o = \sqrt{D_o^2 - H^2} \tag{8}$$

$$\sin E_o = H/D_o \tag{9}$$

$$\cos A_o = \pm \frac{\sqrt{D_m^2 - H^2}}{R_o} \tag{10}$$

$$\dot{D}_o = s^2\,T_o/D_o \tag{11}$$

$$\dot{E}_o = \frac{s^2\,H\,T_o \cos A_o}{D_o^2 \sqrt{D_m^2 - H^2}} \tag{12}$$

$$\dot{A}_o = \frac{s \cos^2 A_o}{\sqrt{D_m^2 - H^2}} \tag{13}$$

Next, the necessary partial derivatives of the predicted position equations:

$$A_p = A_o + \delta_1 \tag{14}$$

$$E_p = E_o + \sigma_1 , \tag{15}$$

where δ_1, σ_1 are the azimuth and elevation leads, and kinematic lead equations (derived from Figure 1):

$$\sin \sigma_1 = \frac{\dot{E}_o D_o t_p}{D_p} - (1 - \cos \delta_1) \sin E_o \cos E_p \tag{16}$$

$$\sin \delta_1 = \frac{\dot{A}_o D_o t_p \cos E_o}{D_p \cos E_p} \tag{17}$$

$$D_p = \frac{D_o + \dot{D}_o t_p}{\cos \sigma_1 - (1 - \cos \delta_1) \cos E_o \cos E_p} \tag{18}$$

$$t_p = F (D_p, E_p) \tag{19}$$

were determined and algebraically solved for $\Delta\delta_1$ and $\Delta\sigma_1$ in terms of the known present position and predicted position data. The algebraic solution results in the following matrix equation:

$$AX = DC \tag{20}$$

where:

$$X = (\Delta\sigma_1,\ \Delta\delta_1,\ \Delta D_p)^t \tag{21}$$

$$C = (\Delta D_o,\ \Delta \dot{D}_o,\ \Delta E_o,\ \Delta \dot{E}_o,\ \Delta \dot{A}_o)^t \tag{22}$$

$$A = (a_{ij})\ i,\ j,\ = 1, 2, 3 \tag{23}$$

with:

$$a_{11} = \cos\sigma_1 - \frac{\dot{E}_o D_o}{D_p}\left(\frac{\partial t_p}{\partial E_p}\right) - (1 - \cos\delta_1)\sin E_o \sin E_p \tag{24}$$

$$a_{21} = -\frac{\dot{A}_o D_o \cos E_o}{D_p}\left(\frac{\partial t_p}{\partial E_p}\right) - \sin\delta_1 \sin E_p \tag{25}$$

$$a_{31} = (1 - \cos\delta_1)\cos E_o \sin E_p - \frac{\dot{D}_o}{D_p}\left(\frac{\partial t_p}{\partial E_p}\right)\sin\delta_1 \tag{26}$$

$$a_{12} = \sin\delta_1 \sin E_o \cos E_p \tag{27}$$

$$a_{22} = \cos E_p \cos\delta_1 \tag{28}$$

$$a_{32} = -\sin\delta_1 \cos E_o \cos E_p \tag{29}$$

$$a_{13} = \frac{\dot{E}_o D_o}{D_p}\left(\frac{t_p}{D_p} - \frac{\partial t_p}{\partial D_p}\right) \tag{30}$$

$$a_{23} = \frac{1}{D_p}\sin\delta_1 \cos E_p - \frac{\dot{A}_o D_o \cos E_o}{D_p}\left(\frac{\partial t_p}{\partial D_p}\right) \tag{31}$$

$$a_{33} = \frac{1}{D_p}\left[\cos\delta_1 - (1-\cos\delta_1)\cos E_o \cos E_p - \dot{D}_o \frac{\partial t_p}{\partial D_p}\right] \quad (32)$$

and $D = (d_{ij})$; $i = 1, 2, 3$; $j = 1, 2, 3, 4, 5$;

with:

$$d_{11} = \frac{\dot{E}_o t_p}{D_p} \quad (33)$$

$$d_{12} = 0 \quad (34)$$

$$d_{13} = \frac{\dot{E}_o D_o}{D_p}\left(\frac{\partial t_p}{\partial E_p}\right) - (1-\cos\delta_1)\cos(E_p + E_o) \quad (35)$$

$$d_{14} = \frac{D_o}{D_p} \quad (36)$$

$$d_{15} = 0 \quad (37)$$

$$d_{21} = \frac{\dot{A}_o t_p}{D_p} \cdot \cos E_o \quad (38)$$

$$d_{22} = 0 \quad (39)$$

$$d_{23} = \frac{\dot{A}_o D_o}{D_p}\cos E_o\left(\frac{\partial t_p}{\partial E_p}\right) + \sin\delta_1 \sin E_p - \frac{\dot{A}_o D_o t_p}{D_p}\sin E_o \quad (40)$$

$$d_{24} = 0 \quad (41)$$

$$d_{25} = \frac{D_o t_p}{D_p}\cos E_o \quad (42)$$

$$d_{31} = \frac{1}{D_p} \tag{43}$$

$$d_{32} = \frac{t_p}{D_p} \tag{44}$$

$$d_{33} = \frac{D_o}{D_p} \left(\frac{\partial t_p}{\partial E_p} \right) - (1 - \cos \delta_1) \sin (E_p + E_o) \tag{45}$$

$$d_{34} = 0 \tag{46}$$

$$d_{35} = 0 \tag{47}$$

Next, solve the matrix equations for X. Observe that X, in particular $\Delta \delta_1$ and $\Delta \sigma_1$, is determined for points along a given target path and errors in present position data. The equations

$$\Delta A_p = \Delta A_o + \Delta \delta_1 \tag{48}$$

$$\Delta E_p = \Delta E_o + \Delta \sigma_1 \tag{49}$$

are now used to compute the errors in predicted position azimuth and elevation. Finally, MD is computed by the equation

$$MD = D_p \left[\left(\Delta A_p \cos E_p \right)^2 + \left(\Delta E_p \right)^2 \right]^{1/2} \tag{50}$$

Observe that MD is the distance between P_p and the point where the projectile pierces the plane drawn through P_p and perpendicular to line GP_p. Although MD is not the minimum distance between projectile and target, it is an excellent approximation to the correct answer.

F015B is the FORTRAN program of the above miss distance method. To repeat, it provides MD tables and discrete graphs as a function of present position errors for equal increments in T_o, D_o and D_p for the XM246 ballistics, XM220 ballistics, and Oerlikon ballistics.

MISS DISTANCE AS A FUNCTION OF GUN PLATFORM OUT-OF-LEVEL

In the previous miss distance analysis, the mathematical development assumed that the gun platform (or deck plane) was level. Now assume that the deck plane makes an angle (D) with the level plane and that the line MN, the intersection of these planes, makes an angle (A) with the cross-over line projected into the level plane. Although the angles A and D completely describe the gun platform out-of-level condition, many people prefer to use the cant of the trunnion (C_g) and the weapon bore tilt (T_g) in such an analysis. The FORTRAN program (page 15) includes both out-of-level descriptions.

Figure 2 pictures that portion of a sphere which represents the antiaircraft problem with respect to the gun pivot point (G). Observe that the figure depicts the level plane of the earth in a neighborhood of G, the deck plane of the gun, the angles A, D, C_g, T_g and the projections onto the spheres surface of the predicted point, weapon bore line point, etc. The spherical triangles so formed suggest the following mathematical technique to determine miss distance.

For given values of A and D and any predicted point (D_p, A_p, E_p), compute the cant of the trunnion, the weapon bore tilt, and the miss distance due to the given out-of-level condition. The right spherical triangles MN_4M_1 and PN_2N_3 produce the equations:

$$\tan C_g = \tan D \cos (A - A_p) \tag{51}$$

$$\tan \Delta A'_p = \tan E_p \tan C_g \tag{52}$$

which permit one to compute C_g and $\Delta A'_p$ under the restriction that $\Delta A'_p \geq 0$. To compute T_g and E', use the right spherical triangles

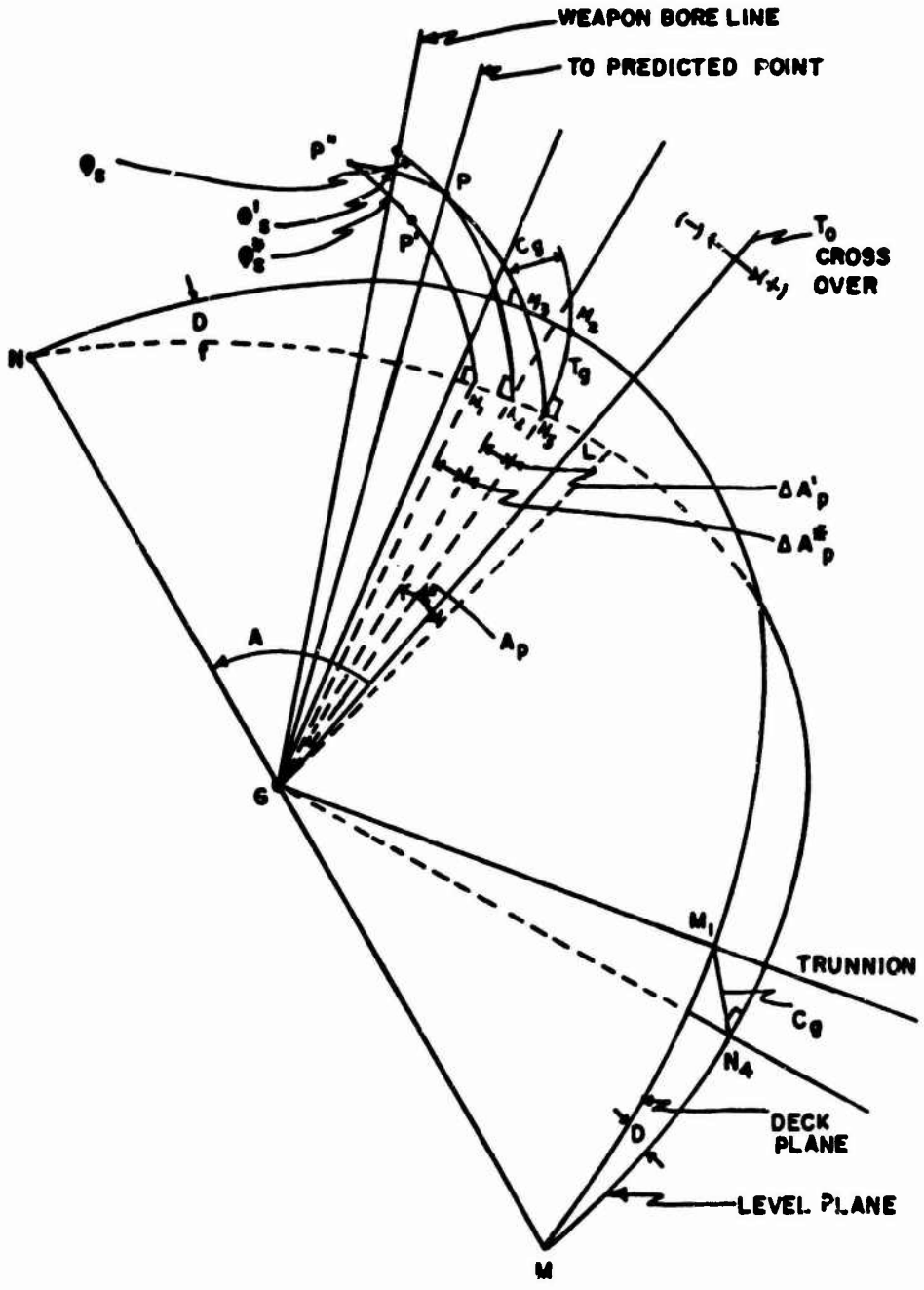


Figure 2. Platform Out-of-Level Fire Control Prediction Geometry of Level Flight Constant Velocity Targets

$N N_3 M_2$ and $N N_3 M_3$ to obtain:

$$\tan T_g = \sin \left[\left| A - (A_p + \Delta A'_p) \right| \right] \tan D \qquad (53)$$

$$\sin E' = \sin D \sin \left[\left| A - (A_p + \Delta A'_p) \right| \right] \qquad (54)$$

Observe that C_g and T_g have now been computed.

To compute the projectile-target miss distance determined by the out-of-level condition given by A and D or C_g and T_g, proceed as follows. In the right spherical triangle PN_2N_3, determine E'_p by first computing $E'_p + E'$ using equations:

$$\cos (E'_p + E') = \cos \Delta A'_p \cos E_p \qquad (55)$$

$$E'_p = (E'_p + E') - E' \qquad (56)$$

Since superelevation φ_s is a function of D_p and E_p, that is, $\varphi_s = f(D_p, E_p)$, the superelevation φ'_s relative to the deck plane may be computed using the given predicted slant range D_p and the predicted elevation angle sensed by the gun, denoted E'_p. Observe that E'_p is the elevation of point P relative to the deck plane. Along the E'_p arc, the superelevation φ'_s is appended to obtain the projected point P''. The line GP'' (not shown in the figure) represents the gun line due to the out-of-level condition determined by A and D.

Since the drop of a projectile is normal to the level plane (ground plane), the spherical right triangle $P''N_1N_2$ is used to derive the equation:

$$\sin \varphi^* = \cos C_g \sin (\varphi'_s + E'_p + E') \qquad (57)$$

which is used to compute the quadrant elevation φ^* of the point P''. For the ballistics under consideration, superelevation may be expressed

as a second degree polynomial in predicted slant range and predicted elevation. Using this polynomial equation in the form

$$\varphi_s^* = f(D_p, \varphi^* - \varphi_s^*) \tag{58}$$

one may solve for φ_s^*, the superelevation associated with the point P'. The angle E_p^*, the elevation of P' with respect to the level plane, is computed from

$$E_p^* = \varphi^* - \varphi_s^* \quad . \tag{59}$$

Next, use the right spherical triangle $P''N_1N_2$ to determine the equation

$$\tan(\Delta A_p^* + \Delta A_p') = \sin C_g \tan(\varphi_s' + E_p' + E'). \tag{60}$$

After computing $\Delta A_p^* + \Delta A_p'$, solve for ΔA_p^* using the equation.

$$\Delta A_p^* = \left|(\Delta A_p^* + \Delta A_p') - \Delta A_p\right| \tag{61}$$

The azimuth of point P' is found by solving

$$A_p^* = A_p - \Delta A_p^* \tag{62}$$

Assuming that the sphere has radius D_p, compute the miss distance (which is approximately equal to the distance from P to P') by the formula:

$$MD = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{63}$$

where:

$$x_1 = D_p \cos E_p \cos A_p \qquad x_2 = D_p \cos E_p^* \cos A_p^*$$
$$y_1 = D_p \cos E_p \sin A_p \qquad y_2 = D_p \cos E_p^* \sin A_p^*$$
$$z_1 = D_p \sin E_p \qquad z_2 = D_p \sin E_p^* \tag{64}$$

MISS DISTANCE AS A FUNCTION OF BALLISTIC EFFECTS

Since ballistic effects (air density change, muzzle velocity change, etc.) produce a change in t_p, the corresponding change made in A_p and E_p can be computed. That is, for each point on a given target path, compute its corresponding δ_1 and σ_1 by using equations (16 and 17). Recall that all of the data concerning points P_p and P_o are known. For a given ballistic effect, Δt_p is known and a new value of δ_1 (call it δ_1^*), corresponding to the ballistic effect is determined by

$$\sin \delta_1^* = \frac{\dot{A}_o D_o (t_p + \Delta t_p) \cos E_o}{D_p \cos E_p} \tag{65}$$

For this ballistic effect, a new value of σ_1 (call it σ_1^*) is found using

$$\sin \sigma_1^* = \frac{\dot{E}_o D_o (t_p + \Delta t_p)}{D_p} - (1 - \cos \delta_1^*) \sin E_o \cos E_p . \tag{66}$$

The errors in A_p and E_p due to this ballistic effect are given by

$$\Delta A_p = (\delta_1^* - \delta_1) \tag{67}$$

$$\Delta E_p = (\sigma_1^* - \sigma_1) \tag{68}$$

and the miss distance is

$$MD = D_p \sqrt{(\Delta A_p \cos E_p)^2 + (\Delta E_p)^2} \tag{69}$$

FORTRAN PROGRAM

Miss Distance as a Function of Unit Input Errors

Program Miss Distance (Primary) is an extended FORTRAN program utilizing the CDC 6600 computer. Input data consists of altitude, speed, and slant cross-over range (which are constant); and maximum predicted slant range, variable to be incremented (i.e., predicted slant range, present position slant range, or time along target path), magnitude of incremental value, and ballistics being used.

The ballistics currently incorporated in the program are: 20 mm XM220, 20 mm XM246, and 35 mm Oerlikon. The program considers unit errors in present position slant range (1 meter); present position slant range rate (1 mil/sec); sightline azimuth (1 mil); sightline azimuth rate (1 mil/sec).

The program considers a unit error in one of the above variables and generates a corresponding miss distance in meters. Since the miss distance is a function of present and predicted slant range, time along target path, and time of flight of projectile, the values are displayed in the output.

The program also considers platform out-of-level conditions and generates a tilt and cant and the miss distance due to tilt and cant.

Tables I, II, and III contain computer program symbology, compilation, and output, respectively, and Figure 3 is a discrete graph of the computer output.

TABLE I.
Miss Distance Computer Program Symbology

	Unit	Symbol
Input		
Height	Meter	H
Speed	Knot	S
Cross-over range	Meter	DM
Predicted slant range (initial)	Meter	DP
Incremental value[a]	Meter or second	T
Dummy variable[b]		LQ
Dummy variable[c]		LW
Deck plane out-of-level	Radian	GD
Angle between cross-over and where level and deck planes intersect	Radian	GA GS
Output		
Present position slant range	Meter	DO
Predicted slant range	Meter	DP
Aircraft time of flight from slant range to cross-over	Second	TO
Projectile time of flight	Second	TTP
Miss distance		
With respect to unit errors in:		
Delta D_o	Meter	DEL DO
Delta $\dot{D}_o$	Meter	DEL DOD
Delta E_o	Meter	DEL EO
Delta $\dot{E}_o$	Meter	DEL EOD
Delta A_o	Meter	DEL AOD
With respect to:		
Tilt and cant	Meter	T + C
Cant of trunnion	Radian	CANT
Weapon bore tilt[d]	Radian	TILT
Output graphs		
T_o (independent variable vs.		
Del D_o (dependent variable)		
Del D_{od} (dependent variable)		
Del E_o (dependent variable)		
Del E_{od} (dependent variable)		
Del A_{od} (dependent variable)		

[a]Present position slant range (D_o), time along target path (T_o), or predicted slant range (D_p) may be evenly incremented. T is the incremental value.

[b]LQ tells program what variable to increment evenly; i.e., LQ = $0/D_o$, $1/T_o$, $2/D_p$.

[c]LW tells program what projectile ballistics are to be used; i.e., LW = 1/20 mm XM246, 2/20 mm XM220, 3/35 mm Oerlikon.

[d]Tilt of deck for a particular point.

TABLE II.

Miss Distance Computer Program as a Function of Present Position Data Errors and Gun Platform Out-of-Level

```
      PROGRAM MISSDIS(INPUT,OUTPUT,TAPE1=INPUT,TAPE3=OUTPUT,TAPE6=OUTPUT
     1)
      DIMENSION AA(3,3),BB(3,5),R(6),Y(5),TR1(200),TR2(200),TR3(200),TR4
     1(200),TR5(200),TIME(200)
    1 FORMAT(1HI,*D A T A    G I V E N    *,*HEIGHT = *,F5.0,5X,*SPEED =
     1 *,F4.0,5X,*CROSSOVER = *,F5.0,5X,*GD = *,F9.6,5X,*GA = *,F9.6)
    2 FORMAT(25X,*(MET)*,15X,*(KTS)*,15X,*(MET)*,13X,*(RAD)*,13X,*(RAD)*
     1)
    3 FORMAT(/,25X,*PROJECTILE  BEING  USED  IS  THE  XM246 - 20MM*)
    4 FORMAT(/,25X,*PROJECTILE  BEING  USED  IS  THE  XM220 - 20MM*)
   41 FORMAT(/,25X,*PROJECTILE  BEING  USED  IS  THE  OERLIKON - 35MM*)
    5 FORMAT (/////,35X,* M I S S  D I S T A N C E  W I T H  R E S P E
     1 C T  TO*)
    6 FORMAT(3X,*DO*,5X,*DP*,7X,*TO*,7X,*TTP*,6X,*DEL DO*,3X,*DEL DOD*,3
     1X,*DEL EO*,3X,*DEL EOD*,3X,*DEL AOD*,5X,*T + C*,5X,*TILT*,7X,*CANT
     2*)
    7 FORMAT(1X,*(MET)*,2X,*(MET)*,4X,*(SEC)*,5X,*(SEC)*,6X,*(MET)*,4X,*
     1(MET)*,5X,*(MET)*,4X,*(MET)*,5X,*(MET)*,6X,*(MET)*,4X,*(RAD)*,6X,*
     2(RAD)*,////)
    8 FORMAT(F6.0,2X,F5.0,2X,F8.3,2X,F7.4,2X,6(F8.4,2X),2(F9.6,2X),//)
    9 FORMAT(5F10.0,2I1)
   10 READ(1,9) H,S,DM,DP,T,LO,LW
      IF(H.EQ.69.) STOP
C HEIGHT,SPEED,CROSSOVER,INITIAL PREDICTED SLANT RANGE
C INCREMENTAL VALUE,TELLS PROGRAM WHAT TO INCREMENT I.E. LO=0/DO,1/TO,2/DP
C PROJECTILE BEING USED LW=1/XM246-20MM,2/XM220-20MM,3/OERLIKON-35MM
      READ(1,9) GD,GA
C WEAPON STATION OUT OF LEVEL CONDITIONS - USED TO GENERATE TILT AND CANT
      WRITE(3,1) H,S,DM,GD,GA
      WRITE(3,2)
      GO TO (101,102,111),LW
  101 WRITE(3,3)
      GO TO 103
  102 WRITE(3,4)
      GO TO 103
  111 WRITE(3,41)
  103 WRITE(3,5)
      WRITE(3,6)
      WRITE(3,7)
      AZ=0.
      WW=0.
      IJK=0
      QP=DP
      S=S*.5144444
      GO TO (11,12,13),LW
   11 A=-.360405178
      B=.263020363E-02
      C=-.318534839E-05
      D=.30504026E-08
      E=-.105461794E-11
      F=.137998534E-15
      GO TO 16
   12 A=-.526557146
      B=.351994444E-02
      C=-.438896456E-05
```

```
      D=.394783459E-08 )
      E=-.13527887E-11 )
      F=.174334879E-15 \
      GO TO 16
   13 A=.124671731E-02
      B=.840470153E-03
      C=.139856858E-06
      D=-.195025201E-10
      E=.669250001E-14
      F=-.331920159E-18
   16 TTP=A+B*DP+C*DP*DP+D*DP**3+E*DP**4+F*DP**5
      TT=A+B*DM+C*DM*DM+D*DM**3+E*DM**4+F*DM**5
      DD=SQRT(DM*DM+S*S*TT*TT)
      TP=-SQRT(DP*DP-DM*DM)/S
      TD=TP-TTP
      XMIN=TD
      PTP=-TP
      PTD=PTP-TTP
      XMAX=PTD
      CP=SQRT(DM*DM+S*S*PTD*PTD)
      DD=SQRT(DM*DM+S*S*TD*TD)
      IF(LD.EQ.0 ) GO TO 116
      GO TO (17,19),LD
  116 IDD=DD/100.
      DD=IDD*100.
      TD=-SQRT(DD*DD-DM*DM)/S
      CALL RANGE(AZ,DP,A,B,C,D,E,F,WW,DM,S,TD,DD,DP,TP,TTP)
      GO TO 19
   17 ITD=TD
      TD=ITD
      DD=SQRT(DM*DM+S*S*TD*TD)
      CALL RANGE(AZ,DP,A,B,C,D,E,F,WW,DM,S,TD,DD,DP,TP,TTP)
   19 FDP=B+2.*C*DP+3.*D*DP*DP+4.*E*DP**3+5.*F*DP**4
      FEP=0.
      IJK=IJK+1
      TIME(IJK)=TD
      RM=SQRT(DM*DM-H*H)
      RP=SQRT(DP*DP-H*H)
      RD=SQRT(DD*DD-H*H)
      EP=ASIN(H/DP)
      ED=ACOS(RD/DD)
      AP=-ACOS(RM/RP)
      IF(AZ.EQ.1.) AP=-AP
      AD=-ACOS(RM/RD)
      IF(AZ.EQ.-1.) AD=-AD
      AOD=(S*COS(AD)*COS(AD))/RM
      DOD=(S*S*TD)/DD
      EDD=-(S*S*H*TD*COS(AD))/(RM*DD*DD)
      DELI=AP-AD
      SIGI=EP-ED
      AA(1,1)=COS(SIGI)-(EDD*DD*FEP)/DP-(1.-COS(DELI))*SIN(ED)*SIN(EP)
      AA(2,1)=(-AOD*DD*COS(ED)*FEP)/DP-SIN(DELI)*SIN(EP)
      AA(3,1)=(1.-COS(DELI))*COS(ED)*SIN(EP)-DOD*FEP/DP-SIN(SIGI)
      AA(1,2)=SIN(DELI)*SIN(ED)*COS(EP)
      AA(2,2)=COS(EP)*COS(DELI)
```

```
      AA(3,2)=-SIN(DEL1)*COS(ED)*COS(EP)
      AA(1,3)=EDD*(DO/DP)*(TTP/DP-FDP)
      AA(2,3)=SIN(DEL1)*COS(EP)/DP-ADD*(DD/DP)*COS(EO)*FDP
      AA(3,3)=(COS(SIG1)-(1.-COS(DEL1))*COS(ED)*COS(EP)-DOD*FDP)/DP
      BB(1,1)=EOD*TTP/DP
      BB(2,1)=ADD*TTP*COS(EO)/DP
      BB(3,1)=1./DP
      BB(1,2)=0.
      BB(2,2)=0.
      BB(3,2)=TTP/DP
      BB(1,3)=(EOD*DO*FEP)/DP-(1.-COS(DEL1))*COS(EP+ED)
      BB(2,3)=(AOD*DD*COS(EO)*FEP)/DP+SIN(DEL1)*SIN(EP)-AOD*(DO/DP)*TTP*
     1SIN(ED)
      BB(3,3)=DO*FEP/DP-(1.-COS(DEL1))*SIN(EP+ED)
      BB(1,4)=DD/DP
      BB(2,4)=0.
      BB(3,4)=0.
      BB(1,5)=0.
      BB(2,5)=DD*TTP*COS(ED)/DP
      BB(3,5)=0.
      M=0
   20 DO 21 K=1,5
   21 Y(K)=0.0
      Y(M+1)=1.
      RD=.001
      Y(3)=Y(3)*RD
      Y(4)=Y(4)*RD
      Y(5)=Y(5)*RD
C INPUT ERRORS GIVEN AS UNIT ONE - TO CHANGE VALUE OF INPUT ERRORS
C CHANGE Y(M+1)=X
C  Y(1)=DDO,Y(2)=DDDO,Y(3)=DED,Y(4)=DEDD,Y(5)=DADD
C  Y(1)=MET,Y(2)=MET,Y(3)=MIL,Y(4)=MIL,Y(5)=MIL
      M=M+1
      DAO=0.
      Y1=BB(1,1)*Y(1)+BB(1,2)*Y(2)+BB(1,3)*Y(3)+BB(1,4)*Y(4)+BB(1,5)*Y(5
     1)
      Y2=BB(2,1)*Y(1)+BB(2,2)*Y(2)+BB(2,3)*Y(3)+BB(2,4)*Y(4)+BB(2,5)*Y(5
     1)
      Y3=BB(3,1)*Y(1)+BB(3,2)*Y(2)+BB(3,3)*Y(3)+BB(3,4)*Y(4)+BB(3,5)*Y(5
     1)
      Q=AA(1,1)*(AA(2,2)*AA(3,3)-AA(2,3)*AA(3,2))-AA(1,2)*(AA(2,1)*AA(3,
     13)-AA(2,3)*AA(3,1))+AA(1,3)*(AA(2,1)*AA(3,2)-AA(2,2)*AA(3,1))
C X1=DELTA SIGMA(1) AND X2=DELTA(1)
      X1=(Y1*(AA(2,2)*AA(3,3)-AA(2,3)*AA(3,2))-AA(1,2)*(Y2*AA(3,3)-AA(2,
     13)*Y3)+AA(1,3)*(Y2*AA(3,2)-AA(2,2)*Y3))/Q
      X2=(AA(1,1)*(Y2*AA(3,3)-AA(2,3)*Y3)-Y1*(AA(2,1)*AA(3,3)-AA(2,3)*AA
     1(3,1))+AA(1,3)*(AA(2,1)*Y3-Y2*AA(3,1)))/Q
      DAP=DAO+X2
      DEP=Y(3)+X1
      P=DP*SQRT((DAP*COS(EP))**2+DEP*DEP)
      IF(Y(1).EQ.1.) KK=0
      KK=KK+1.
      R(KK)=P
      IF(KK.NE.5) GO TO 20
      IF(GO.EQ.0.0) GO TO 22
```

```
      CALL TILT (GD,GA,AP,EP,DP,LW,CG,TG,R6),RETURNS(23)
   22 TG=0.0
      CG=0.0
      R(6)=0.0
      GO TO 122
   23 R(6)=R6
  122 WRITE(3,8) DO,DP,TO,TTP,(R(J),J=1,6),TG,CG
      TR1(IJK)=R(1)
      TR2(IJK)=R(2)
      TR3(IJK)=R(3)
      TR4(IJK)=R(4)
      TR5(IJK)=R(5)
      IF(IJK.EQ.1) YMAX=R(4)
      IF(LQ.EQ.0.) GO TO 123
      GO TO(24,25),LQ
  123 IF(WW.EQ.1.) GO TO 28
      DO=DO-T
      IF(DO.LT.DD) AZ=1.
      IF(DO.LT.DM) GO TO 27
      TO=-SQRT(DO*DO-DM*DM)/S
      CALL RANGE(AZ,QP,A,B,C,D,E,F,WW,DM,S,TO,DO,DP,TP,TTP)
      GO TO 19
   24 TO=TO+T
      DO=SQRT(DM*DM+S*S*TO*TO)
      IF(DO.LT.DD) AZ=1.
      IF(TO.GE.0.) WW=1.
      IF(WW.EQ.1.) GO TO 29
      GO TO 30
   29 IF(DO.GT.CP) GO TO 31
   30 CALL RANGE(AZ,QP,A,B,C,D,E,F,WW,DM,S,TO,DO,DP,TP,TTP)
      GO TO 19
   25 IF(AZ.EQ.1) GO TO 26
      DP=DP-T
      IF(DP.GE.DM) GO TO 127
      AZ=1.
   26 DP=DP+T
  127 TTP=A+B*DP+C*DP*DP+D*DP**3+E*DP**4+F*DP**5
      TP=-SQRT(DP*DP-DM*DM)/S
      IF(AZ.EQ.1.) TP=-TP
      TO=TP-TTP
      DO=SQRT(DM*DM+S*S*TO*TO)
       IF(AZ.NE.1.) GO TO 19
      IF(DO.GT.CP) GO TO 31
      GO TO 19
   27 WW=1.
   28 DO=DO+T
      IF(DO.GT.CP) GO TO 31
      TO=SQRT(DO*DO-DM*DM)/S
      CALL RANGE(AZ,QP,A,B,C,D,E,F,WW,DM,S,TO,DO,DP,TP,TTP)
      GO TO 19
   31 CALLNANCYL(4HTIME,5H(SEC),9HMISS DIST,5H(MET))
      CALL NANCY5(TIME,TR1,IJK,TIME,TR2,IJK,TIME,TR3,IJK,TIME,TR4,IJK,TI
     1ME,TR5,IJK,XMAX,XMIN,YMAX,0.,1,1)
      GO TO 10
      END
```

```
      SUBROUTINE TILT (GD,GA,AP,EP,DP,LW,CG,TG,R),RETURNS(X1)
      A=GA
      D=GD
      CG=ATAN(TAN(D)*COS(A-AP))
      DAPP=ATAN(TAN(EP)*TAN(CG))
      BG=ABS(A-AP-DAPP)
      TG=ATAN(SIN(BG)*TAN(D))
      PE=ASIN(SIN(D)*SIN(BG))
      EPP=ACOS(COS(DAPP)*COS(EP))-PE
      GO TO (201,202),LW
  201 A1=-.141015931E+01
      A2=.116460351E-01
      A3=-.930808903E-05
      A4=.774244660E-08
      A5=-.764050957E-12
      AA=.100826567E+01
      BB=-.199919697E-04
      CC=-.431901515E-06
      GO TO 203
  202 A1=-.213948141E+01
      A2=.139796979E-01
      A3=-.1317609296-04
      A4=.967310434E-08
      A5=-.107226552E-11
      AA=.102274E+01
      BB=-.48951057E-04
      CC=-.42547514E-06
  203 EEP=EPP*1000.
      AK=A1+A2*DP+A3*DP*DP+A4*DP**3+A5*DP**4
      BK=AA+BB*EEP+CC*EEP*EEP
      CK=AK*(2.*CC*EEP+BB)-1.
      PHISP=(-CK-SQRT(CK*CK-4.*AK*AK*BK*CC))/(2.*AK*CC)
      PHISP=PHISP/1000.
      PHISTAR=ASIN(COS(CG)*SIN(PHISP+EPP+PE))
      PHISST=AK*(AA+BB*PHISTAR+CC*PHISTAR*PHISTAR)
      PHISST=PHISST/1000.
      EPST=PHISTAR-PHISST
      DAPST=ATAN(SIN(CG)*TAN(PHISP+EPP+PE))-PE
      APST=AP-DAPST
      X=DP*COS(EP)*COS(AP)
      Y=DP*COS(EP)*SIN(AP)
      Z=DP*SIN(EP)
      XP=DP*COS(EPST)*COS(APST)
      YP=DP*COS(EPST)*SIN(APST)
      ZP=DP*SIN(EPST)
      R=SQRT((X-XP)**2+(Y-YP)**2+(Z-ZP)**2)
      RETURN X1
      END
```

```
      SUBROUTINE RANGE(AZ,QP,A,B,C,D,E,F,WW,DM,S,TO,DO,PD,PT,PTT)
      PDMAX=DO
      PDMIN=DM
      IF(AZ.EQ.1.) PDMAX=QP
      IF(AZ.EQ.I.) PDMIN=DM
      IF(WW.EQ.I) PDMIN=DO
      PD=PDMAX
  300 PT=-SQRT(PD*PD-DM*DM)/S
      IF(AZ.EQ.1.) PT=-PT
      PTT=A+B*PD+C*PD*PD+D*PD**3+E*PD**4+F*PD**5
      OT=PT-PTT
      G=TO-OT
      IF(ABS(G).LT..00I) GO TO 304
      IF(AZ.EQ.I) GO TO 30I
      IF(G)302,304,303
  301 IF(G)303,304,302
  302 PDMIN=PD
      PD=(PD+PDMAX)/2.
      GO TO 300
  303 PDMAX=PD
      PD=(PD+PDMIN)/2.
      GO TO 300
  304 RETURN
      END
```

TABLE III.

Miss Distance Computer Output For Unit Input Errors

Data given: Height, 1000 meters; Speed, 400 kts;
Cross-over, 1200 meters; GD, -0.000000 radian;
GA, -0.000000 radian;
Projectile used, 35 mm Oerlikon

				MISS DISTANCE WITH RESPECT TO							
DO (MET)	DP (MET)	TO (SEC)	TTP (SEC)	DEL DO (MET)	DEL DOO (MET)	DEL EO (MET)	DEL EOO (MET)	DEL AOO (MET)	Y-C (MET)	TILT (RAD)	CANT (RAD)
2932.	2421.	-13.000	2.7015	.1036	.0045	2.6175	2.9453	7.6047	0.0000	0.000000	0.000000
2838.	2351.	-12.500	2.6771	.1056	.0415	2.3478	2.8517	7.1292	0.0000	0.000000	0.000000
2745.	2281.	-12.000	2.5747	.1078	.0587	2.2768	2.7589	6.6006	0.0000	0.000000	0.000000
2643.	2211.	-11.500	2.4751	.1102	.0561	2.2073	2.6667	6.1000	0.0000	0.000000	0.000000
2562.	2142.	-11.000	2.3775	.1127	.0535	2.1388	2.5754	5.6244	0.0000	0.000000	0.000000
2472.	2074.	-10.500	2.2825	.1155	.0512	2.0694	2.4850	5.1750	0.0000	0.000000	0.000000
2382.	2006.	-10.000	2.1899	.1185	.0489	2.0015	2.3956	4.7504	0.0000	0.000000	0.000000
2294.	1939.	-9.500	2.0999	.1218	.0468	1.9346	2.3074	4.3506	0.0000	0.000000	0.000000
2207.	1873.	-9.000	2.0123	.1253	.0447	1.8683	2.2204	3.9739	0.0000	0.000000	0.000000
2121.	1806.	-8.500	1.9268	.1292	.0429	1.8035	2.1348	3.6216	0.0000	0.000000	0.000000
2037.	1744.	-8.000	1.8441	.1333	.0411	1.7398	2.0508	3.2914	0.0000	0.000000	0.000000
1955.	1683.	-7.500	1.7638	.1379	.0395	1.6779	1.9686	2.9843	0.0000	0.000000	0.000000
1875.	1623.	-7.000	1.6925	.1429	.0380	1.6176	1.8885	2.6996	0.0000	0.000000	0.000000
1797.	1565.	-6.500	1.6210	.1483	.0367	1.5596	1.8106	2.4346	0.0000	0.000000	0.000000
1722.	1509.	-6.000	1.5531	.1541	.0355	1.5040	1.7352	2.1914	0.0000	0.000000	0.000000
1650.	1456.	-5.500	1.4896	.1604	.0345	1.4514	1.6628	1.9680	0.0000	0.000000	0.000000

1581.	1407.	-5.000	1.4306	.1672	.0335	1.4021	1.5037	1.7062	0.0000	0.000000	0.000000
1516.	1361.	-4.500	1.3767	.1745	.0329	1.3568	1.5202	1.5836	0.0000	0.000000	0.000000
1455.	1320.	-4.000	1.3203	.1822	.0323	1.3162	1.4670	1.6198	0.0000	0.000000	0.000000
1400.	1284.	-3.500	1.2863	.1904	.0321	1.2813	1.4184	1.2756	0.0000	0.000000	0.000000
1349.	1253.	-3.000	1.2509	.1989	.0320	1.2527	1.3591	1.1503	0.0000	0.000000	0.000000
1306.	1220.	-2.500	1.2230	.2077	.0323	1.2317	1.3136	1.0443	0.0000	0.000000	0.000000
1269.	1211.	-2.000	1.2032	.2162	.0325	1.2194	1.2747	.9579	0.0000	0.000000	0.000000
1239.	1202.	-1.500	1.1924	.2247	.0334	1.2166	1.2431	.8922	0.0000	0.000000	0.000000
1210.	1201.	-1.000	1.1913	.2330	.0348	1.2235	1.2195	.8483	0.0000	0.000000	0.000000
1204.	1204.	-.500	1.2003	.2404	.0365	1.2391	1.2050	.8277	0.0000	0.000000	0.000000
1200.	1220.	0.000	1.2202	.2465	.0383	1.2615	1.2000	.8322	0.0000	0.000000	0.000000
1204.	1253.	.500	1.2510	.0730	.1847	1.2221	1.2056	.8536	0.0000	0.000000	0.000000
1210.	1269.	1.000	1.2929	.1403	.0219	1.1403	1.2264	.8892	0.0000	0.000000	0.000000
1239.	1335.	1.500	1.3603	.3200	.0634	1.0531	1.2693	.9736	0.0000	0.000000	0.000000
1269.	1390.	2.000	1.4109	.5004	.0955	.9710	1.3372	.8544	0.0000	0.000000	0.000000
1306.	1453.	2.500	1.4801	.6714	1.1473	.9274	1.4304	.7965	0.0000	0.000000	0.000000
1349.	1525.	3.000	1.5722	.8129	1.3855	.9388	1.5472	.6755	0.0000	0.000000	0.000000
1400.	1603.	3.500	1.6640	.9250	1.6083	1.0047	1.6849	.4672	0.0000	0.000000	0.000000
1455.	1689.	4.000	1.7755	.9986	1.7960	1.1186	1.8394	.1709	0.0000	0.000000	0.000000
1516.	1781.	4.500	1.8924	1.0203	1.9321	1.2386	2.0044	.3675	0.0000	0.000000	0.000000
1581.	1878.	5.000	2.0190	1.0167	2.0853	1.3744	2.1691	.0112	0.0000	0.000000	0.000000
1650.	1980.	5.500	2.1557	.9739	2.0217	1.5183	2.3210	1.3034	0.0000	0.000000	0.000000
1722.	2088.	6.000	2.3020	.9130	1.9958	1.6437	2.4542	1.9919	0.0000	0.000000	0.000000
1797.	2200.	6.500	2.4567	.8452	1.9451	1.7757	2.5648	2.6244	0.0000	0.000000	0.000000
1875.	2316.	7.000	2.6200	.7780	1.8835	1.9076	2.6540	3.2705	0.0000	0.000000	0.000000
1955.	2437.	7.500	2.8040	.7151	1.8200	2.0414	2.7290	3.9602	0.0000	0.000000	0.000000

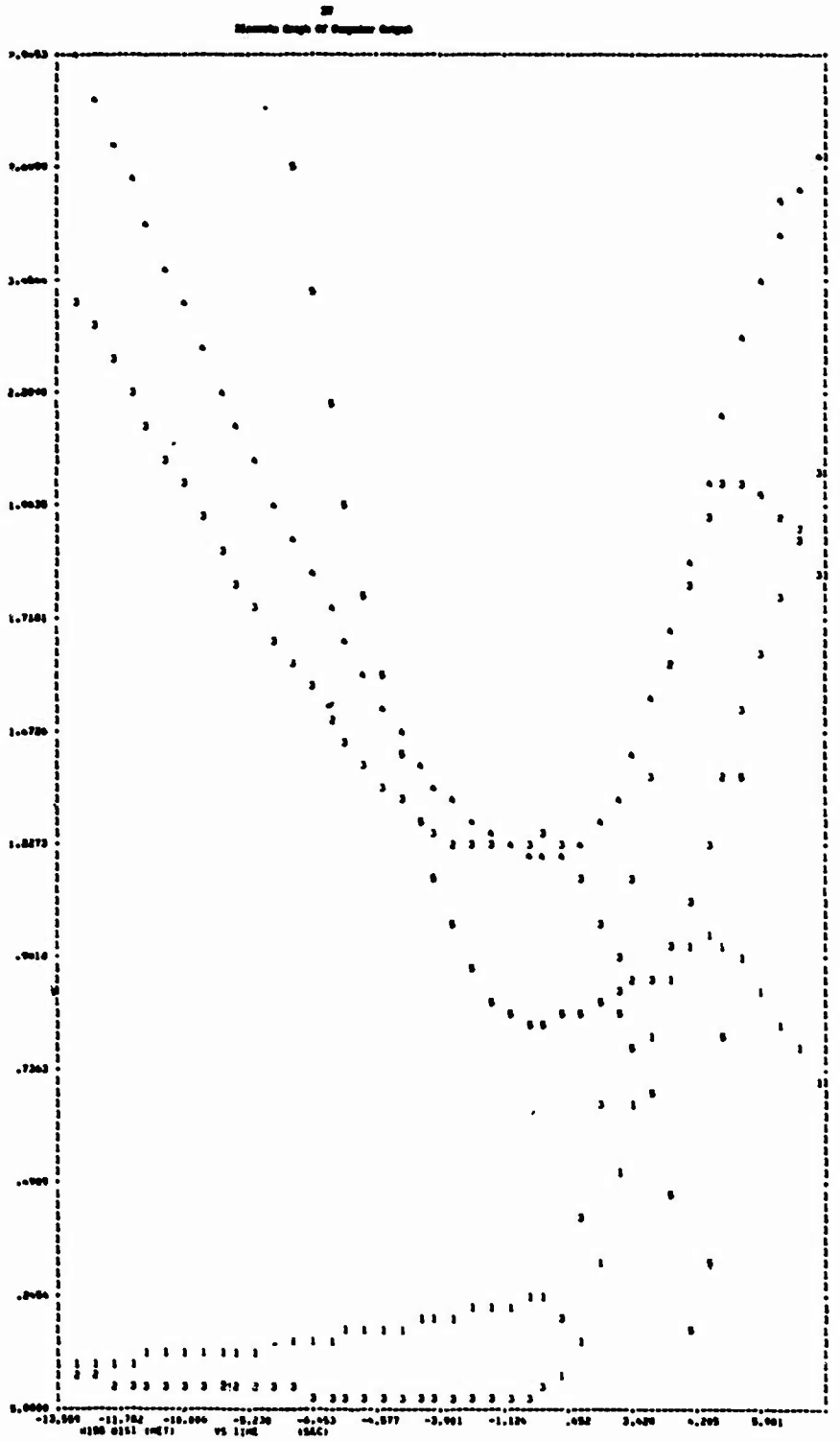


Figure 3. Discrete Graph of Computer Output

FORTRAN PROGRAM

Miss Distance as a Function of Ballistic Effects

Program Miss Distance (secondary) is an extended FORTRAN program utilizing the CDC 6600 computer. Input data consists of altitude, speed, and slant cross-over range (which are constant); and maximum predicted slant range, percent error in time of flight, and ballistics being used.

The ballistics currently incorporated in the program are: 20 mm XM220, 20 mm XM246, and 35 mm Oerlikon. The program considers a fixed (percentage) error in time of flight which is a function of ballistic effects and which generates a corresponding miss distance in meters. Since the miss distance is a function of present and predicted slant range, time along target path and time of flight of projectile, their values are displayed in the output.

Tables IV and V contain computer program compilation and output, respectively.

TABLE IV.
Miss Distance Computer Program as A Function of Ballistic Effects

```
      PROGRAM MDE(INPUT,OUTPUT,TAPE1=INPUT,TAPE3=OUTPUT)
    1 FORMAT(5F10.0,I1)
    2 FORMAT(1H1,*D A T A    G I V E N    *,*HEIGHT = *,F5.0,5X,*SPEED =
     1 *,F4.0,5X,*CROSSOVER = *,F5.0)
    3 FORMAT(25X,*(MET)*,15X,*(KTS)*,15X,*(MET)*)
    4 FORMAT(/,25X,*PROJECTILE  BEING  USED  IS  THE  XM246 - 20MM*)
    5 FORMAT(/,25X,*PROJECTILE  BEING  USED  IS  THE  XM220 - 20MM*)
   51 FORMAT(/,25X,*PROJECTILE  BEING  USED  IS  THE  OERLIKON - 35MM*)
    6 FORMAT(/,25X,*ASSUMING BALLISTIC  EFFECTS, TTP CHANGE = *,F4.2,1X,
     1*PERCENT*)
    7 FORMAT(////,3X,*DP*,5X,*DO*,7X,*TO*,7X,*TTP*,8X,*MISS DISTANCE*)
    8 FORMAT(1X,*(MET)*,2X,*(MET)*,4X,*(SEC)*,5X,*(SEC)*,9X,*(MET)*,//)
    9 FORMAT(F6.0,2X,F5.0,2X,F8.3,2X,F7.4,9X,F7.4,/)
      READ(1,1) H,S,DM,DP,CTTP,LW
      Q=100.*CTTP
      WRITE(3,2) H,S,DM
      WRITE(3,3)
      S=S*.5144444
      GO TO (10,11,21),LW
   10 WRITE(3,4)
      GO TO 12
   11 WRITE(3,5)
      GO TO 12
   21 WRITE(3,51)
   12 WRITE(3,6)Q
      WRITE(3,7)
      WRITE(3,8)
      GO TO (13,14,16),LW
   13 A=-.360405178
      B=.283020363E-02
      C=-.3185348 9E-05
      D=.30504026E-08
      E=-.1054 1794E-11
      F=.137998534E-15
      GO TO 15
   14 A=-.526557146
      B=.351994444E-02
      C=-.438896456E-05
      D=.394783459E-08
      E=-.13527887E-11
      F=.174334879E-15
      GO TO 15
   16 A=.124671731E-02
      B=.840470153E-03
      C=.139856858E-06
      D=-.195025201E-10
      E=.669250001E-14
      F=-.331920159E-18
   15 TTP=A+B*DP+C*DP*DP+D*DP**3+E*DP**4+F*DP**5
      DTTP=CTTP*TTP
      TP=-SQRT(DP*DP-DM*DM)/S
      TO=TP-TTP
      DO=SQRT(DM*DM+S*S*TO*TO)
      RM=SQRT(DM*DM-H*H)
      RP=SQRT(DP*DP-H*H)
```

```
RO=SQRT(DD*DD-H*H)
AP=-ACOS(RM/RP)
AD=-ACOS(RM/RO)
EP=ASIN(H/DP)
EO=ACOS(RO/DO)
ADD=(S*COS(AD)*COS(AD))/RM
EDD=-(S*S*H*TO*COS(AD))/(RM*DO*DD)
DEL1=AP-AD
SIG1=EP-ED
DEL1S=ASIN(ADD*DD*(TTP+DTTP)*COS(ED)/(DP*COS(EP)))
SIG1S=ASIN(EDD*DO *TTP+DTTP)/DP-(1.-COS(DEL1S))*SIN(EO)*COS(EP))
DAP=DEL1S-DEL1
DEP=SIG1S-SIG1
DX=DP*(SQRT((DAP*COS(EP))**2+DEP*DEP))
WRITE(3,9)DP,DD,TD,TTP,DX
DP=DP-100.
IF(DP.LT.DM)STOP
GO TO 15
END
```

TABLE V.

Miss Distance Computer Output For
Ballistic Effects

Data given: Height, 1000 meters; Speed, 400 kts;
Cross-over, 1200 meters;
Projectile used, 35 mm Oerlikon.

Assuming ballistic effects, ttp change = 1.00 percent.

MISS DISTANCE WITH RESPECT TO				
DP (MET)	DO (MET)	TO (SEC)	TTP (SEC)	MISS DISTANCE (MET)
2500.	3037.	-13.559	2.9008	2.3421
2400.	2904.	-12.851	2.7500	2.3204
2300.	2771.	-12.138	2.6028	2.2994
2200.	2639.	-11.420	2.4592	2.2791
2100.	2506.	-10.694	2.3190	2.2594
2000.	2375.	-9.957	2.1821	2.2404
1900.	2243.	-9.207	2.0483	2.2221
1800.	2111.	-8.437	1.9175	2.2045
1700.	1978.	-7.641	1.7896	2.1875
1600.	1845.	-6.807	1.6645	2.1713
1500.	1709.	-5.916	1.5422	2.1560
1400.	1571.	-4.927	1.4224	2.1427
1300.	1425.	-3.735	1.3053	2.1407
1200.	1225.	-1.191	1.1906	2.4516

CONCLUSIONS

The FORTRAN programs developed in this report provide a useable method of evaluating the effectiveness of a gun air defense system. The projectile-target miss distance (defined on page 9) evaluated by the programs is tabulated and graphed as a function of errors in:

1. Slant range,
2. Slant range rate,
3. Azimuth,
4. Azimuth rate,
5. Elevation,
6. Elevation rate,
7. Projectile time of flight,
8. Gun platform out-of-level

for targets having level flight, constant target speed, and any ballistic data.

These FORTRAN programs apply to any gun air defense system as they are based on a theoretical study of the problem. Thus, a gun system's component errors are not an input to these programs.

APPENDIX A

DETERMINATION OF MISS DISTANCE TABLES FOR EQUAL INCREMENTS OF D_o, T_o, or D_p

The Miss Distance program provides miss distance tables as a function of present position errors for increments in D_o, T_o, or D_p. To increment D_o, T_o, or D_p, use the procedure described below.

1. Incremental Procedure

a. If it is desired to increment D_o, assign to variable T the value of the increment and set dummy variable LQ equal to 0. Program proceeds to take D_o to the next lower hundredth (i.e., D_o = 3722 implies 3700) and calculates T_o for this value of D_o; then calls subroutine RANGE, which uses an iterative process to generate D_p.

b. If it is desired to increment T_o, assign to variable T the value of the increment and set dummy variable LQ equal to 1. Program proceeds to take T_o to the next higher integer (i.e., T_o = - 7.35 implies -7.00) and calculates D_o for this value of T_o; then calls subroutine RANGE, which uses an iterative process to generate D_p.

c. If it is desired to increment D_p, assign to variable T the value of the increment and set dummy variable LQ equal to 2. Program proceeds to automatically increment D_p.

2. Subroutine range RANGE

The iterative method employed is the bisection method. Suppose the limits of D_p (i.e., $D_m \leq D_p \leq D_o$ before crossover) are known. Initially set $D_p = D_o$ and generate T_o' and compare it with T_o. If $T_o - T_o'$ is negative, set D_p = previous D_p + D_o divided by 2 and set previous $D_m \leq D_p \leq D_o$. If $T_o - T_o'$ is positive, set D_p = previous D_p + D_m divided by 2 and set previous $D_m \leq D_p \leq$ previous D_p. Continue to generate T_o' until $T_o - T_o'$ reaches the desired accuracy.

APPENDIX B

LEAST SQUARE POLYNOMIAL "FAN" FIT TO DATA OF THREE VARIABLES

For data involving three variables, say x, y, and z, a least square polynomial fit can be quickly and easily found when the data describes a fan (Figure B-1) in two dimensions.

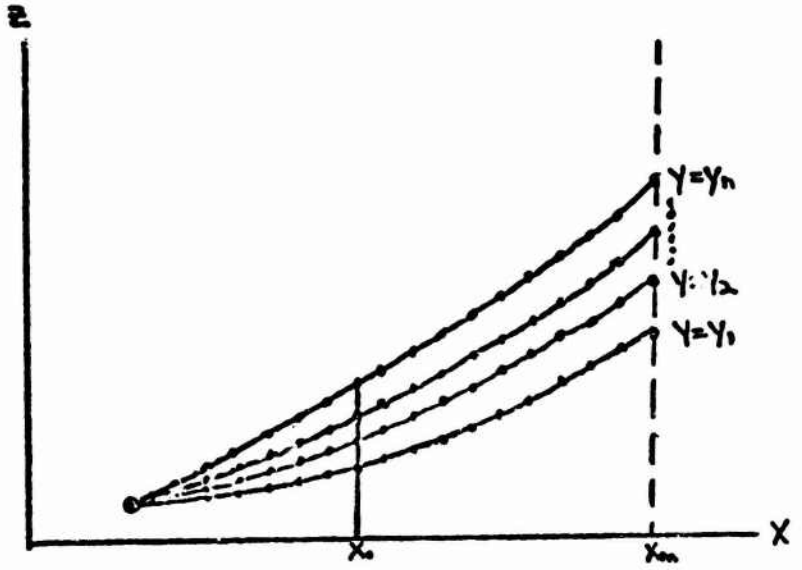


Figure B-1. Graph of Discrete Data for Functions Possessing a "Fan" Shape

Observe that the curves $y_1, y_2, \ldots, y_n$ spread out like a fan. Although the curves need not be straight lines, it is assumed that each curve has the same degree polynomial fit as a function of x in domain defined by table.

Using the tabular data for $y = y_1$, the smallest value of y, determine a least square polynomial fit $z = P_1(x, y_1)$. In like manner determine $z = P_n(x, y_n)$, where y_n is the largest value of y in table.

Let x_m be the value of x that maximizes $P_n(x, y_n) - P_1(x, y_1)$. Now graph z as a function of y (Figure B-2) when $x = x_m$ and determine

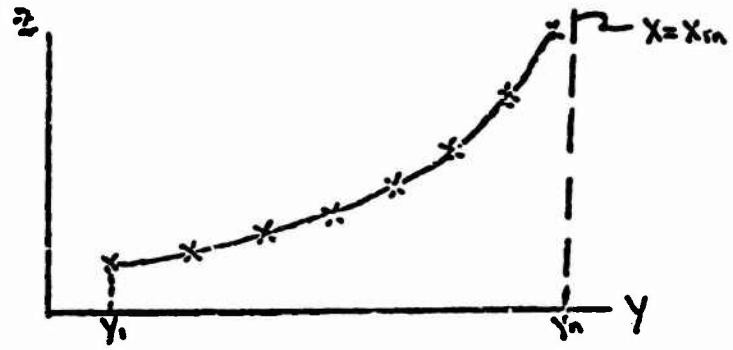


Figure B-2. Graph of Discrete Data for $z = f(y)$

the least square polynomial fit $z = Q(x_m, y)$. Because the curves of Figure B-1 spread out uniformly, determine the least square polynomial fit to the three variables using the proportion

$$\frac{P(x, y) - P_1(x, y_1)}{P_n(x, y_n) - P_1(x, y_1)} = \frac{P(x_m, y) - P_1(x_m, y_1)}{P_n(x_m, y_n) - P_1(x_m, y_1)}$$

Solving for P(x, y), we obtain

$$z = P(x, y) = \frac{P(x_m, y) - P_1(x_m, y_1)}{P_n(x_m, y_n) - P_1(x_m, y)} \left(P_n(x, y_n) - P_1(x, y_1)\right) + P_1(x, y_1)$$

the desired least square polynomial fit to the given tabular data in three variables.

ACKNOWLEDGEMENT: The author wishes to thank Alice Jones and Samuel Yosen for writing and testing the FORTRAN programs which print miss distance tables and discrete graphs as functions of errors in tracking data, gun-platform out-of-level and errors in projectile time of flight.

COMPUTER-GENERATION OF CIRCULAR GRAPHICAL FIRING SCALES

Diana Dadamo, Joseph Kaszupski
Fire Control Development and Engineering Directorate
U.S. Army Frankford Arsenal
Philadelphia, Pa. 19137

ABSTRACT. The use of firing tables for the prediction of mortar fire is well-established; a four-inch circular firing scale was developed to provide a convenient and inexpensive adaption of these tables for use in the field. The generation of this scale was done originally by hand, but because the nature of these scales required a significant revision of the layout for a minor change in the data, a more practical approach (i.e., computer-generated) was attempted. This paper explains the development and content of the computer program and a description of the scale while providing an explicit comparison between hand drawn and computer-generated plots.

1. INTRODUCTION. The use of tabular firing tables for prediction of mortar fire is an established practice; in the field, the graphical firing fan remains the basic form of calculation although there have been advances in automating this process. The linear representation of the firing tables was a major step in presenting the tabular firing tables in a more convenient form for use in Fire Direction Center. The computer has been utilized to provide advances for this grid method of calculation without disturbing the general form of operation as is now present in the mortar team.

Two separate efforts to computerize the paraphernalia of the FDC are as follows: computer generation of linear scales for use with the graphical firing fan, and a computer-generated disc that would replace the linear scale (in some situations). The manual creation of masters for geometric firing scales is tedious and involves human error; generation of these scales by computer eliminates most of this error and also provides a means to easily produce a new scale due to modification of ballistic data. Since the data for the scale is produced by a computer, program out efforts are ultimately intended to be used as an alternate form of output for the ballistic programs.

Generation of the linear scale by computer for the 4.2 inch mortar was done successfully by Frances Edelman of Frankford Arsenal, who showed that computer generation of linear scales could be done, within the necessary tolerances. Subsequently, Frankford is now working on a more general plotting program for the generation of all linear scales presently in use, which provides for direct modification of the scale.

The original use of the graphical firing scales was for the prediction of artillery fire; the adaption of these graphical firing fans for the mortar is actually not ideal because of the mobility of that weapon. With the development of the new lightweight company mortar, it became essential to provide an extremely portable representation of the ballistic data. Studies in the reduction of tabular data resulted in the development of a compact circular firing scale.

It is a four inch disc consisting of an outer range scale and internal curves which each represent a separate charge of the ammunition. These curves are used as base lines to display data of elevation, maximum ordinate of the trajectory at certain elevations and time of flight of the projectile. A cursor is used to align the desired range on the outer range scale with the elevation of these charges which have that range capability. Above the elevations, there is miscellaneous data that is needed for illumination or smoke rounds.

These curves are functions of the maximum and minimum ranges of the charges, and although they are not ballistically representative (but are constructed with a human engineering viewpoint) they are constructed uniquely for each set of data so that the maximum amount of space inside the range scale may be used for the plotting. A change in the maximum range of the weapon or certain charges would obviously require a complete new scale. Hand plotting of a scale that would be invalidated by such a minor change in data is impractical.

2. EXPLANATION OF THE COMPUTER PROGRAM

A computer-generated plotting program for the circular firing scale has been developed at Frankford which produces a plot directly from ballistical data.

The program operates in the batch mode under two options externally controlled by a switch; a more complete program that calculates the constants for a set of data or tests if modified data is compatible with a previous run; and also produces the equations and points used in the plotting of the curves themselves; and an actual plotting program. The use of an external indicator for the choice of run eliminates the need for program modification as a result of a change in data.

An important calculation is the angle increments needed to correlate the angle of the range and the angle needed to locate the point at which the cursor positioned at that range will intersect the curves. The position of the cursor, however, is not from the center of the disc but is constructed so that it is parallel to the Y-axis at some point in its rotation, with the displacement on the X-axis from the center equal to the minimum plotting radius. The angle formed by the intersection of the cursor and a radius of the curve defines a constant angle differential for each curve--which is then used to locate and position the tic marks.

In order to use a maximum amount of the space within the center of the disc, the curves are constructed so that the cursor and at least one curve intersect the inner circle at the 800 mil elevation point; and the minimum radius of the innermost charge is a specified distance from the center of the disc. Each curve starts at a different angle; the range is a function of the angle of rotation from this starting angle. The latter maintains a constant distance between any two curves which is a desirable quality from a human engineering viewpoint.

According to a report on the French Curve Program developed by Lemont Blake, the appearance of a point-to-point plotted curve will be smooth if the increment used in the plotting of the curve is .1 the curvature of the curve. This fact is used in the calculation of the points for the curve on which the points are to be plotted, keeping in mind the resolution of the plotter and the fact that these curves do not affect the accuracy of the data plotted on them.

Generation of the curves and the previous constants are done in the calculation part of the program; the plotting program requires this data before the plotting of the actual ballistic data may be done. Each charge is plotted separately to minimize the amount of core-memory needed for the program and also to produce a tentative output even if one charge's data is inaccurately read in. The plotting program tests for readability for every point that is plotted on the curves.

At parts on the curves, the increments between the points to be plotted may be below the comfortable resolution of the eye. At every interval of 100 mils on the lower scale that marks elevation, tests are made of the distances between the calculated positions of data in 10 mil increments. If any of these distances between the tic marks is below a minimum value, the values of the elevations at those points that are not multiples of 50 will not be plotted. Instead, the distance between the two 50 mil increments in that interval will be tested. If those points can be plotted with the given resolutions, they will be; if not, that interval will be omitted from the plotting array and the next 100 mil interval will be examined. At these points which have been accepted for actual plotting, tic marks are drawn at the angle at which the cursor passes through the curve.

Next the data of time of flight and maximum ordinate are examined. These points will be plotted on the upper scale if they are sufficiently distant from each other; initially the time of flight points take precedence over the M.O. points and are formed into arrays with the maxium ordinate points interspaced, if they meet the resolution test. Some further conventions have been established for the plotting at these points concerning the priority so that the density of the maximum ordinate and TOF points are comparative. However, there is more freedom with the plotting of the maximum ordinate points since the numbers labeling these points are encased with a box to distinguish them from the time of flight points. The point on the curve which the maximum ordinate represents is indicated by the apex of a triangle so that the box may be shifted one-half its length in either direction if that is necessary to meet the resolution test.

Use of these tests will produce a readable scale for each charge. While the calculation program tests for possible interaction between the curves, certain sets of data may produce interference at some points. Also, the appearance of a scale may be improved by the manipulation of some points that do not follow the patterns of the program. For this purpose, an editing program that exists in a timeshare environment was designed (which is based on the program used in the generation of the linear scale). This program allows the user to change the position of points from the array of points to be plotted, after an initial run of the batch-program.

This provides a quick method of editing a plot that could still be computer generated. The existence of a fiber-optics light head for plotters could be used, finally, to make a photo master directly, thus eliminating the intermediate steps in the present method.

Generation of these scales by computer would then eliminate much of the errors and reduce the time involved in the creation of masters, and would provide a general ability for the generation of future scales.

A COMPUTATIONAL SYSTEM FOR NUMERICAL INTEGRATION WITH RIGOROUS ERROR ESTIMATION

Julia H. Gray and L. B. Rall

Mathematics Research Center

University of Wisconsin-Madison

ABSTRACT

By use of the concepts of interval analysis, it is possible to construct methods for numerical integration which makes possible rigorous estimates of errors due to imprecise data, round-off, and truncation, and also makes it possible to detect the use of incorrect formulas. On the basis of parameters obtained from interval integration formulas, the calculation of numerical integrals can be optimized with respect to time and accuracy. It is also often possible to improve the accuracy by intersection of interval results. This paper describes a computational system of this type which has been implemented as a computer program for the UNIVAC 1108/1110, using available software for interval analysis and automatic differentiation.

1. Numerical integration. One of the classical problems of numerical analysis is to obtain accurate values of definite integrals of the form

$$z = \int_a^b y(x)dx \tag{1.1}$$

in case the integration cannot be carried out explicitly. Methods for this process of numerical integration (or quadrature) have been developed since the first days of the calculus, and there is a vast literature devoted to them. A treatment of the basic concepts of numerical integration may be found in the book by Davis and Rabinowitz (1967), as well as in standard texts on numerical analysis, such as the ones by Milne (1949) or Mysovskih (1969).

In what follows, it will be assumed that the limits of integration a, b in (1.1) are finite, and that y(x) is a Riemann integrable function (Davis and Rabinowitz (1967), pp. 4-6). If the numerical integration method under discussion explicitly requires derivatives of y(x) of certain orders, then these will be assumed to exist without further ado.

Sponsored by the U. S. Army under Contract No. DA-31-124-ARO-D-462 and by National Science Foundation Grant No. GP-40381.

Attention will be devoted for the most part to the use of expressions of the form

$$\int_a^b y(x) = \sum_{i=1}^{n} y(x_i)w_i + e(y) \tag{1.2}$$

for the numerical integration. In (1.2), the term

$$r(y) = \sum_{i=1}^{n} y(x_i)w_i \tag{1.3}$$

is called the rule of numerical integration, it being assumed that the nodes x_i are such that $a \leq x_1 < x_2 < \ldots < x_n \leq b$, and the weights $w_1, w_2, \ldots, w_n$ are given. For example, the choice for $n = 3$ of

$$\begin{cases} x_1 = a, \quad x_2 = \dfrac{a+b}{2}, \quad x_3 = b, \\ \\ w_1 = \dfrac{b-a}{6}, \quad w_2 = \dfrac{2(b-a)}{3}, \quad w_3 = \dfrac{b-a}{6}, \end{cases} \tag{1.4}$$

gives the familiar and useful Simpson's rule.

Assuming that all indicated calculations can be performed exactly, the term $e(y)$ in (1.2) is the error in taking

$$z_r = r(y) = \sum_{i=1}^{n} y(x_i)w_i \tag{1.5}$$

as an approximation to the true value z of the integral (1.1). Consequently, $e(y)$ is frequently called the (truncation) error term in (1.2). For most of the numerical integration methods used in practice, expressions for $e(y)$ may be given in terms of derivatives of y. In the case of Simpson's rule (1.4), one has

$$e(y) = -\frac{(b-a)^5}{2880} y^{iv}(\xi), \tag{1.6}$$

where ξ is an (unknown) point in the open interval $a < \xi < b$, and it is assumed that the fourth derivative $y^{iv}(x)$ of $y(x)$ is continuous (Davis and Rabinowitz (1967), p. 19).

The operator f defined by

$$f(y) = r(y) + e(y)$$

is called an integration formula for the functions in its domain. A typical example is Simpson's formula

$$(1.8) \qquad f(y) = \frac{b-a}{6}\left[y(a) + 4y\left(\frac{a+b}{2}\right) + y(b)\right] - \frac{(b-a)^5}{2880}\, y^{iv}(\xi) \ ,$$

obtained from (1.4) and (1.6). The methods of numerical integration to be considered below are based for the most part on the use of integration formulas, the truncation error term being brought into the computation along with the rule.

2. Error estimation. In actual calculation of integrals, one obtains an approximation z^* to the true value z of the integral (1.1). The error, or difference between z and z^*, arises from one or more of the following sources:

(i) Imprecise data. The function $y(x)$ to be integrated depends on coefficients which can be specified to a certain precision, as, for example, the results of measurements.

(ii) Round-off error. This occurs in almost every calculation because only a finite number of digits can be used. In fact, one has to deal with this type of error even if the integral is defined explicitly. In most computing machines, one has round-off in the conversion of decimal numbers to and from binary, from the fact that many constants, such as $\frac{1}{3}$, π, the nodes and weights of Gaussian integration rules (Milne (1949), pp. 285-288), etc., cannot be represented exactly as machine numbers, and in the arithmetic operations required in the numerical evaluation of the integral.

(iii) Truncation error. This type of error is introduced by the neglect or approximation of the term $e(y)$ in the integration formula (1.7), as in (1.5).

(iv) Incorrect integration formula. In theory, an integration formula f gives the exact value

$$(2.1) \qquad z = f(y) = \int_a^b y(x)dx$$

of the integral (1.1) of each function $y(x)$ for which it is defined. The use of an incorrect formula, while rare, has been observed in practice.

While it is usually impossible to eliminate errors of types (i)-(iii) completely, one would hope to be able to minimize their effect on the accuracy of the final results, and to obtain reliable estimates of the resulting error. Errors of type (iv) are best described as blunders, and can be

avoided by careful checking to see if the integration formula is correctly derived, and is suitable for application to the function being integrated. In a properly designed computational system, it is often possible to detect errors of type (iv).

The approach to error estimation taken here will be rigorous, that is, the computation will yield a positive number ε such that the inequality

$$|z - z^*| \leq \varepsilon \tag{2.2}$$

is guaranteed to hold. Inequality (2.2) is, of course, a bound for the absolute error of z^* as an approximation to z. In certain applications, one may want a positive bound ρ for the relative error, that is,

$$\left|\frac{z - z^*}{z}\right| \leq \rho , \tag{2.3}$$

it being assumed that $zz^* > 0$. Rigorous error bounds are usually considered to be costly in terms of analytical effort, and to yield pessimistic results in that the bounds obtained are often much larger than the actual errors in test cases. The amount of computational time devoted to obtaining the error bound may also greatly exceed the time required to calculate the approximation z^*. The system described here estimates the effects of errors of types (i)-(iii) automatically, and obtains z^* and ε (or ρ) as the results of the same computation. Experience indicates that the error bounds given are usually fairly realistic. The machine time (and storage) required, however, is large compared to straightforward evaluation of numerical integration rules without error estimation, so certain features for optimization have been included, and information relative to cost is provided.

As an alternative to rigorous error estimation, there are ways to obtain indicative error estimates, which are easy to compute and are ordinarily of the same order of magnitude as the actual errors. For example, one may note the dependence of the truncation error term $e(y)$ on the length of the interval of integration, as in (1.6), and deduce an estimate for the error based on comparison of values obtained from the integration rule for two or more different subdivisions of the total interval of integration (see Noble (1964), pp. 231-237). While useful in practice, this type of error estimation still requires additional machine time, and does not provide the rigorously guaranteed estimates of error which may be required in certain applications.

3. Interval analysis. The system for numerical integration presented here is based on the computation of a (closed) interval $Z = [c, d]$ which is known to contain the value z of the integral (1.1). If $z \in Z$, then one may take the midpoint $\mu(Z)$ of Z,

(3.1) $$z^* = \mu(z) = \frac{c+d}{2}$$

as an approximation to z, with absolute error bounded by

(3.2) $$\varepsilon = \frac{1}{2}(d - c) = \frac{1}{2}\,\delta(Z) \quad ,$$

where $\delta(Z) = d - c$ denotes the <u>width</u> of the interval Z.

If one is concerned with relative (or percentage) error, then one may take the <u>harmonic point</u> $\nu(Z)$ of Z,

(3.3) $$z^* = \nu(Z) = \frac{2cd}{c+d}$$

as the corresponding approximation to z, provided that $0 \notin Z$, with the relative error bounded by

(3.4) $$\rho = \left|\frac{d-c}{c+d}\right| = \frac{1}{|c+d|}\,\delta(z) \quad ,$$

and thus 100ρ bounds the percentage error.

The methods of <u>interval analysis</u> (Moore, 1966) will be used to obtain the required intervals. The basic ideas needed for this purpose are those of <u>interval extensions</u> of real numbers and function. An interval $X = [a, b]$ is said to be an <u>extension</u> of the real number x if $x \in X$. In dealing with the theoretical foundations of interval analysis, one may identify real numbers x with the degenerate interval $[x,x]$. In actual computation, however, only a finite set of numbers (the so-called <u>machine numbers</u>) are available, and one deals with non-degenerate extensions. For example, if machine numbers consist of five decimal digits, then the number $\frac{1}{3}$ would have to be represented by an interval extension such as $[0.33333, 0.33334]$. In this case, the given extension is <u>minimal</u>, as it is contained in all other interval extensions of $\frac{1}{3}$ on the same machine.

In the same way, a real function $\phi(x_1, x_2, \ldots, x_n)$ of n real variables $x_1, x_2, \ldots, x_n$ can be extended to an <u>interval function</u> $\Phi(X_1, X_2, \ldots, X_n)$ of n intervals $X_1, X_2, \ldots, X_n$. The requirement that Φ be an extension of ϕ is, of course,

(3.5) $$\phi(x_1, x_2, \ldots, x_n) \in \Phi(X_1, X_2, \ldots, X_n)$$

provided $x_i \in X_i$, $i = 1, 2, \ldots, n$. The rules for mathematical operations on intervals may be derived from this definition. For example,

$$X + Z = [a, b] + [c, d] = [a+c, b+d] \tag{3.6}$$

is the (minimal) interval extension of the function $\phi(x, z) = x + z$.

The use of interval analysis in error estimation is immediate. For example, suppose that the integrand in (1.1) contains the polynomial

$$p(x) = 0.20x^2 + 1.33x - 4.69 \ , \tag{3.7}$$

in which the coefficients are known to be accurate only to two decimal places. The effects of uncertainty in these coefficients may be taken into account by computing with the interval extension

$$P(X) = [0.195, 0.205]X^2 + [1.325, 1.335]X - [4.685, 4.695] \tag{3.8}$$

of (3.7).

Round-off error can also be taken into account by interval analysis. For example, in (3.6), even though a, b, c, d are machine numbers, this may not be true of one or both of the sums $a+c$, $b+d$. By proper rounding, however, one may compute an interval extension of a given function which always contains the true value. In order to minimize round-off error, the computations should be programmed so that the extensions obtained are as close to minimal as possible. Software for the automatic implementation of calculation of interval extensions of arithmetic operations and a number of frequently encountered functions is available for the UNIVAC 1108/1110 (Ladner and Yohe, 1970). It is also important to insure that the conversion between the decimal and binary number systems during input and output also lead to correct interval extensions of the quantities involved (Binstock, Hawkes, and Hsu, 1973).

4. Interval integrals. In order to calculate an interval Z which contains the value z of a given integral (1.1), one may use Riemann sums (Rall, 1965) or Riemann-Stieltjes sums (Decell and Lea, 1966). These methods of numerical integration tend to be time-consuming, and turn out to be special cases of the idea of interval integrals, introduced by R. E. Moore (1965, pp. 76-88; 1966, Chap. 8). For example, the interval version of (1.1) corresponding to the use of Riemann sums is

$$Z = \sum_{i=1}^{n} Y(X_i)\, \delta(X_i) \ , \tag{4.1}$$

where

$$[a,b] \subset \bigcup_{i=1}^{n} X_i \tag{4.2}$$

and $Y(X)$ is an interval extension of $y(x)$ (As a notational convention, a capital letter will denote an interval extension of the quantity indicated by the corresponding lower case symbol.) The interval Z given by (4.1) contains the upper and lower Riemann sums for the integral (1.1), and hence its value z.

More generaliy, one may obtain an interval containing (1.1) by simply calculating an interval extension

$$F(Y) = R(Y) + E(Y) \tag{4.3}$$

of any integration formula (1.7) which is valid for the given integrand. For example, Newton's three-eights formula is

$$\int_{x_0}^{x_3} y(x)dx = \frac{3h}{8}(y_0 + 3y_1 + 3y_2 + y_3) - \frac{3h^5}{80} y^{iv}(\xi) \ , \tag{4.4}$$

(Milne (1949), p. 123), where

$$x_i = x_0 + ih, \quad y_i = y(x_i), \qquad i = 0, 1, 2, 3 \ , \tag{4.5}$$

and $x_0 < \xi < x_3$, assuming that the function $y(x)$ has a continuous fourth derivative $y^{iv}(x)$. The interval version of (4.4) is

$$\int_{x_0}^{x_3} y(x)dx \in \frac{3H}{8}(Y_0 + 3Y_1 + 3Y_2 + Y_3) - \frac{3H^5}{80} Y^{iv}(X) \ , \tag{4.6}$$

where

$$Y_i = Y(X_i) \ , \qquad i = 0, 1, 2, 3 \ , \tag{4.7}$$

and

(4.8) $$[x_0, x_3] \subset X$$

In order to apply the methods of interval analysis to the computation of (4.6), it appears that the program must be given the formula for $y^{iv}(x)$ in addition to the integrand $y(x)$. However, it is possible to program the differentiation of functions by computers, based on a philosophy very similar to translation of formulas into machine language routines (see Moore (1965), pp. 103-112). A number of programs of this type are available, including two for the UNIVAC 1108/1110, one based on the generation of code lists as described by Gray and Reiter (1968), and another which is based on the recursive generation of Taylor coefficients (Knapp and Wanner, 1970).

In case that the required derivative does not exist on the interval X of integration, a suitable error message will be generated by the program for interval evaluation (Gray and Reiter, 1968). This assists in the detection of errors of type (iv), that is, the use of inappropriate or incorrect integration formulas.

5. Optimization. The goal of optimization of the performance of a numerical integration program may be to obtain a result of specified accuracy in minimum time, or else to attain the highest feasible accuracy. A common approach may be made to these problems. It will be assumed that the integration formula (4.3) to be used has the form

(5.1) $$R(Y) = \sum_{i=1}^{n} Y(X_i)\, W_i \ ,$$

corresponding to (1.5), and that the truncation error term has the form

(5.2) $$E(Y) = C \cdot H^{k+1} \cdot Y^{(k)}(X) \ ,$$

where k is a positive integer and C is an (interval) constant. The absolute error of the approximation

(5.3) $$z^* = \mu[F(Y)]$$

will be bounded by

(5.4) $$\varepsilon = \tfrac{1}{2}\delta[F(Y)] = \tfrac{1}{2}\delta[R(Y)] + \tfrac{1}{2}\delta[E(Y)] \ .$$

(Only optimization with respect to absolute error will be considered here; similar results for the percentage error follow from (3.4).)

Experience has shown that $\delta[R(Y)]$, which is proportional to a weighted average of the quantities $\delta[Y(X_i)]$, tends to be relatively constant for a fairly wide range of choices of rules for numerical integration. This quantity thus sets an effective lower bound for the error estimate ε. Thus, one is lead to consider varying the quantity $\delta[E(Y)]$ as the means to optimize the error bound (5.4). This can be done in one of two ways, either by subdividing the interval of integration and applying the given integration formula to each subinterval, or by choosing a different formula which will alter one or both of the values k, C in (5.2). In the first case, suppose that the interval of integration X is divided into ℓ subintervals $X_{01}, X_{12}, \ldots, X_{\ell-1,\ell}$, each of width $\delta(X)/\ell$. This corresponds to replacing H in (5.2) by H/L, and one obtains

$$E(Y) = \left(\frac{1}{L}\right)^{k+1} \cdot C \cdot H^{k+1} \sum_{i=1}^{\ell} Y^{(k)}(X_{i-1,i}) \quad . \tag{5.5}$$

The coefficient $\left(\frac{1}{L}\right)^{k+1}$ in this expression may be made as small as desired simply by taking L large enough. As $C \cdot H^{k+1}$ is constant, the behavior of $\delta[E(Y)]$ can be determined if one can estimate the quantity

$$\delta(\Sigma) = \delta\left[\sum_{i=1}^{\ell} Y^{(k)}(X_{i-1,i})\right] = \sum_{i=1}^{\ell} \delta[Y^{(k)}(X_{i-1,i})] \quad . \tag{5.6}$$

In theory, there are two extreme cases, assuming that successive subintervals $X_{i-1,i}$ and $X_{i,i+1}$ have only an endpoint in common. First, if $y^{(k)}(x)$ is monotone, then

$$\sum_{i=1}^{\ell} \delta[Y^{(k)}(X_{i-1,i})] = \delta[Y^{(k)}(X)] \quad , \tag{5.7}$$

at least approximately, as the maximum (or minimum) of y(x) in the subinterval $X_{i-1,i}$ is its minimum (or maximum) in the subsequent subinterval $X_{i,i+1}$. On the other hand, if $y^{(k)}(x)$ is oscillatory, and attains its maximum and minimum in X in each of the subintervals $X_{i-1,i}$, then

$$\delta[Y^{(k)}(X_{i-1,i})] = \delta[Y^{(k)}(X)], \quad i = 1, 2, \ldots, \ell \quad , \tag{5.8}$$

and thus

$$\sum_{i=1}^{\ell} \delta[Y^{(k)}(X_{i-1,i})] = \ell\, \delta[Y^{(k)}(X)] \quad . \tag{5.9}$$

Of course, equality holds in general in (5.7) and (5.9) only if $Y^{(k)}(X)$ is the exact interval extension of $y^{(k)}(x)$; however, one always has

$$\sum_{i=1}^{\ell} \delta[Y^{(k)}(X_{i-1,i})] \leqq \ell\, \delta[Y^{(k)}(X)] \quad , \tag{5.10}$$

which will be used for estimation. It follows from (5.5) that

$$\delta[E(Y)] \leqq (\frac{1}{\ell})^k \ \delta[C \cdot H^{k+1} \cdot Y^{(k)}(X)] \quad . \tag{5.11}$$

Thus, in order to estimate the optimum number of times to apply a given numerical integration formula to obtain a specified accuracy ε, the formula may be applied <u>once</u> to the entire interval X to obtain the quantities

$$r = \delta[R(Y)], \quad t = \delta[C \cdot H^{k+1} \cdot Y^{(k)}(X)] \quad . \tag{5.12}$$

If $2\varepsilon - r > 0$, then the smallest integer ℓ such that

$$\ell > \sqrt[k]{\frac{t}{2\varepsilon - r}} \tag{5.13}$$

will give the desired estimate. Also, if θ is the time taken for the single application of the numerical integration formula, then

$$\tau = \ell\, \theta \tag{5.14}$$

estimates the total time required for the entire computation.

To obtain optimal accuracy, the above estimates may be modified slightly to provide the necessary information. Suppose that p is that largest positive integer such that

(5.15) $$10^{-p} > r \ .$$

Setting $2\varepsilon = 10^{-p}$ in (5.13) will then give an estimate for the number of repetitions required to obtain p decimal places of accuracy, that is,

(5.16) $$\varepsilon = 5 \cdot 10^{-p-1}$$

The optimization parameters obtained from a single application of the numerical integration formula agree well with the results of numerical experiments. For example, Newton's three-eights formula (4.6) applied to the example

(5.17) $$z = \int_0^2 (\sqrt{1+4x} + \sin 17x)dx$$

gives $\ell = 54$, $\tau = 3.834$ sec. for optimal accuracy of $\varepsilon = 5 \cdot 10^{-5}$ Actual calculation gives

(5.18) $$F(Y) = 4.442072809 \pm 0.000010789 \ ,$$

with $\varepsilon = 1.08 \cdot 10^{-5}$, $\tau = 3.787$ sec.

The same parameters may be used to compare one integration formula with another. For example, the trapezoidal formula

(5.19) $$\int_{x_0}^{x_1} y(x)dx \in \frac{H}{2}(Y_0 + Y_1) - \frac{H^3}{12} Y''(X)$$

may be applied to example (5.17) to obtain the estimates $\ell = 1980$, $\tau = 65.736$ sec., $\varepsilon = 5 \cdot 10^{-5}$. The results actually obtained are

(5.19) $$F(Y) = 4.4420724215 \pm 0.0000206535 \ ,$$

and $\varepsilon = 2.07 \cdot 10^{-5}$, $\tau = 67.045$ sec. This shows that Newton's three-eights formula would be preferable for this application.

6. Intersection methods. If it is known that $z \in Z_1$ and $z \in Z_2$, then

(6.1) $$z \in Z_1 \cap Z_2$$

This gives a way of improving the accuracy of interval integrals, as

(6.2) $$\delta(Z_1 \cap Z_2) \leq \min\{\delta(Z_1), \delta(Z_2)\}$$

One way to obtain intervals Z_1, Z_2 containing the value of a given integral without much additional computation is based on an observation of Milne (1926) concerning pairs of open and closed Newton-Cotes integration formulas. For example, the use of Simpson's formula twice gives

(6.3) $$\int_{x_0}^{x_4} y(x)dx \in Z_c = \frac{H}{3}(Y_0 + 4Y_1 + 2Y_2 + 4Y_3 + Y_4) - \frac{H^5}{45} Y^{iv}(X_{04}) ,$$

where $[x_i, x_j] \subset X_{ij}$, and the corresponding open formula (Milne (1949), p. 127, formula (2)) is

(6.4) $$\int_{x_0}^{x_4} y(x)dx \in Z_o = \frac{4H}{3}(2Y_1 - Y_2 + 2Y_3) + \frac{14H^5}{45} Y^{iv}(X_{04})$$

The difference in sign tends to separate these intervals. For

(6.5) $$\int_0^2 \sqrt{1+4x}\, dx = \frac{13}{3} ,$$

one has

(6.6) $$Z_c = 4.4139288665 \pm 0.0832954055$$

and

(6.7) $$Z_o = 3.1791572275 \pm 1.1661335535$$

Neither result being very accurate. However,

(6.8) $$Z = Z_c \cap Z_o = 4.337962121 \pm 0.007328660$$

is accurate to almost two decimal places, and was obtained without additional evaluations of the integrand or its derivatives.

Pairs of open and closed integration formulas can be used to construct interval versions of predictor-corrector methods for initial value problems for ordinary differential equations. Repetition of corresponding open and closed numerical integration formulas often leads to the situation $Z_c \subset Z_o$. Consequently, the parameters for the closed formula Z_c are used for optimization.

The use of intersection of interval integrals can also lead to the detection of incorrect integration formulas. For example, formula (1) on p. 127 of Milne (1949) should be

(6.9) $$\int_{x_0}^{x_3} y(x)dx = \frac{3h}{2}(y_1 + y_2) + \frac{3h^3}{4} y''(\xi) \quad .$$

(Davis and Rabinowitz (1967), p. 32). This error was discovered when the incorrect formula produced an interval which was disjoint with one obtained by using the trapezoidal formula.

7. Description of the computer program. A computer program has been written for the UNIVAC 1108/1110 to carry out the numerical integration methods presented in this paper. A detailed description may be found in the report by Gray and Rall (1974). Briefly, provisions are made for operation in interactive or batch mode. The interactive mode may be used from a terminal to calculate a few integrals, investigate optimization, etc. In the batch mode, this program may be used for large scale computations, such as the tabulation of functions defined by definite integrals, and could also be used as a subroutine by another program. Provisions are made in each version for repeated application of formulas, optimization, and intersection of results.

The present program makes the following integration formulas available to the user, where n denotes the number of nodes:

1. Riemann sums.
2. Extended Trapezoidal formula, $2 \leq n \leq 25$.
3. Extended Simpson formula, $3 \leq n \leq 25$.
4. Closed Newton-Cotes formulas, $2 \leq n \leq 9$,
5. Open Newton-Cotes formulas $4 \leq n \leq 10$
6. Gaussian formulas $2 \leq n \leq 10$.

Other formulas may be added as desired.

REFERENCES

1. Binstock, W., Hawkes, J., and Hsu, Nai-Ting, An interval input/output package for the UNIVAC 1108. MRC Technical Summary Report No. 1212, University of Wisconsin-Madison, 1973.

2. Davis, P. J. and Rabinowitz, P., Numerical integration. Blaisdell, Waltham, Mass., 1967.

3. Decell, H. P., Jr. and Lea, R. N., Numerical integration and Riemann-Stieltjes sums. SIAM Review 8 (1966), 196-200.

4. Gray, Julia H. and Rall, L. B., INTE: A UNIVAC 1108/1110 program for numerical integration with rigorous error estimation. MRC Technical Summary Report No. 1428, University of Wisconsin-Madison, 1974.

5. Gray, Julia H. and Reiter, A., CODEX - Compiler of differentiable expressions. MRC Technical Summary Report No. 791, University of Wisconsin-Madison, 1968.

6. Knapp, H. and Wanner, G., LIESE II - A program for ordinary differential equations using Lie-series. MRC Technical Summary Report No. 1008, University of Wisconsin-Madison, 1970.

7. Ladner, T. D. and Yohe, J. M., An interval arithmetic package for the UNIVAC 1108. MRC Technical Summary Report No. 1055. University of Wisconsin-Madison, 1970.

8. Milne , W. E., Numerical integration of ordinary differential equations. Amer. Math. Monthly 33 (1926), 455-460.

9. Milne, W. E., Numerical calculus. Princeton University Press, Princeton, N.J., 1949.

10. Moore, R. E., The automatic analysis and control of error in digital computation based on the use of interval numbers. Error in Digital Computation, Vol. I, ed. by L. B. Rall, pp. 61-130. John Wiley and Sons, New York, 1965.

11. Moore, R. E., Interval analysis, Prentice-Hall, Englewood Cliffs, N.J., 1966.

12. Mysovskih, I. P., Lectures on numerical methods. Tr. by L. B. Rall. Wolters-Noordhoff, Groningen, 1969.

13. Noble, B., Numerical methods II: Differences, integration and differential equations. Oliver and Boyd, London, 1964.

14. Rall, L. B., Numerical integration and the solution of integral equations by the use of Riemann sums. SIAM Review 7 (1965), 55-64.

ON THE EFFECTIVE USE OF A LARGE COMPUTER PROGRAM FOR STRUCTURAL CALCULATIONS

E. Cuthill and P. Matula
Naval Ship Research and Development Center
Bethesda, Maryland

ABSTRACT

The general applicability of the finite element method as a numerical method for solving a wide variety of structural analysis problems has made possible the development of several large computer program systems of wide applicability. One of these, the NASTRAN structural analysis program, and its impact on the effective use of computers for structural analysis is discussed.

Part of the discussion centers on the sense in which a given program system such as NASTRAN permits effective use of computers for structural work. The effective use of such large computer programs relates to many factors such as the applicability to the problem at hand, the ease of use, the soundness of the numerical methods used, the provisions made for error checking, the maintenance, consultation and training services available, the wide user groups who share experiences and costs of further improvements and developments.

A brief discussion of some aspects of the numerical methods used in finite element program is included.

The work sponsored by the Office of the Director of Navy Laboratories (DNL) through the Navy NASTRAN Systems Office, Code 1844, Naval Ship Research & Development Center, was carried out under Task Area ZF 099 01 01, Work Unit 1-1844-007.

A. INTRODUCTION

The conference theme, "Optimal Use of Computers in Army R&D", has been a stimulating one for us. We are responding with some thoughts and observations on "The Effective Use of a Large Computer Program for Structural Calculations." Our presentation is in three parts under the following topics:

The finite element method makes possible computer programs for analysis of structural problems.

Such programs can be used effectively.

Large systems of equations can sometimes be solved efficiently with proper sequencing of equations and unknowns.

The first topic involves definition of the finite element method in a general way. The existence of such a method, plus efficient techniques for solving the large, sparse systems of equations it leads to, has made possible the development of general purpose structural analysis programs such as NASTRAN. Such programs for static and dynamic analysis can be applied to a wide range of structures, to almost every type of construction, and to a great variety of loading conditions.

Secondly, the question "Is it possible to use such large, general purpose programs effectively?" is addressed. We note some of the conditions which, if fulfilled, contribute in vital ways to the effective use of such large, general purpose computer programs.

Finally, at the heart of such programs are time-consuming routines for manipulating large sparse matrices and for solving the large equation systems that arise. One aspect of the solution process, vital if these large equation systems are to be solved effectively, is the sequencing of equations and unknowns.

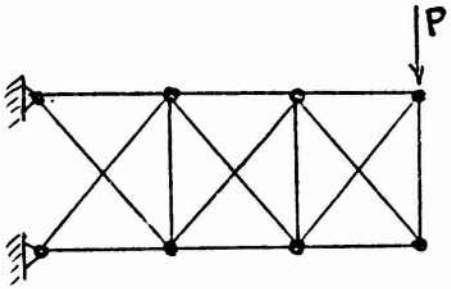
B. THE FINITE ELEMENT METHOD MAKES POSSIBLE COMPUTER PROGRAMS FOR ANALYSIS OF STRUCTURAL PROBLEMS.

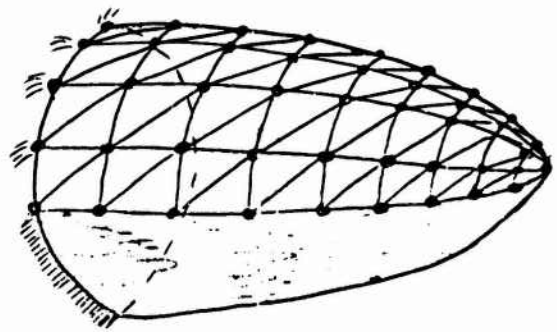
Figure 1 addresses the origin of the finite element method in structural analysis. For example, for each beam element in a structure made up of beams (shown at the left side of the figure), simple beam theory readily provides a force displacement relation. Writing for each joint a force displacement relation that satisfies the boundary conditions generates a set of algebraic equations. Depending on the way they are set up, these equations can be solved for displacements and/or force boundary values and hence for stresses. This suggests that a continuous structure can be modelled with elements assumed connected at a discrete set of points. In this case the function form assumed for the displacements, for example, could be such that, when displacements and their appropriate derivatives are matched at the discrete set of points, appropriate smoothness conditions would be satisfied at all points. Zienkiewicz[1] provides a good description of the finite element method from an engineering point of view.

Figure 2 briefly outlines the finite element method from a mathematical point of view. Mathematically, the finite element method can be interpreted as the Rayleigh-Ritz-Galerkin method in which the trial functions are from a space of piecewise polynomials.[2] For example, for problems involving the beam equations in one dimension, these trial functions could be spline functions. Martin Schultz's recent book entitled "Spline Analyzis"[3] can be interpreted as a book on the finite elements method. A second excellent recent reference is that of Strang and Fix.[2]

---

1. Zienkiewicz, O.C., The Finite Element Method in Engineering Science, 2nd Ed., McGraw-Hill, New York, 1971.
2. Strang, G., and G. J. Fix, An Analysis of the Finite Element Method, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.
3. Schultz, M. H., Spline Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.





- NATURAL STRUCTURE = FINITE ELEMENT MODEL
- ELEMENT DISPL.-FORCE RELATIONSHIP IS READILY AVAILABLE FROM BEAM THEORY.
- FORM A SET OF ALGEBRAIC EQUATIONS SATISFYING BOUNDARY CONDITIONS AT EACH JOINT.
- SOLVE FOR DISPL. AND/OR FORCE BOUNDARY VALUES AND HENCE FOR STRESSES.

- IN FINITE ELEMENT ANALYSIS A CONTINUOUS STRUCTURE IS MODELED WITH ELEMENTS ASSUMED CONNECTED AT DISCRETE POINTS THUS CREATING A SYSTEM WITH A FINITE NUMBER OF DOF UPON WHICH MATRIX ALGEBRA OPERATIONS CAN BE PERFORMED.
- ELEMENTS MUST POSSESS SUCH STIFFNESS CHARACTERISTICS THAT WHEN THE SIZE OF EL'S IS DECREASED, THE STRESSES AND DISPLACEMENTS MUST TEND TO THE EXACT VALUES FOR THE CONTINUOUS SYSTEM.

Figure 1 - Origin of the Finite Element Method

GIVEN THE PROBLEM IN VARIATIONAL FORM;

- SUBDIVIDE THE REGION OF INTEREST INTO SMALLER PIECES OF SIMPLY FORM.
- WITHIN EACH PIECE, USE TRIAL FUNCTIONS SUCH AS LOW ORDER POLYNOMIALS IN SUCH A WAY THAT LOCAL BOUNDARY CONDITIONS ARE IMPOSED.
- DETERMINE THAT COMBINATIONS OF TRIAL FUNCTIONS WHICH IS MINIMIZING.

Figure 2 - The Finite Element Method from a Mathematical Point of View

On the basis of these numerical methods, general computer programs for analyzing wide ranges of structures under general loading conditions have been developed.[4] Major parts of such a finite element program are shown in Figure 3. Of course, this is oversimplified, but it is clear that provision for a wide range of elements and element loading conditions plus a full range of matrix manipulation capabilities can make for a very powerful and general system if a good executive routine is available to facilitate accessing the full range of capabilities.

C. FACTORS THAT INFLUENCE THE EFFECTIVE USE OF LARGE COMPUTER PROGRAMS.

Figure 4 addresses our second topic - "What factors influence the effective use of large computer programs?"

Any large computer program (in a rapidly developing field) requiring several years of development will probably be obsolete in some ways by the time it becomes available. If it is well designed, however, providing modularity and privision for maintenance and upgrading, the threat of obsolescence should be no problem. Modules can be replaced and upgraded as required. When we first expressed an interest in NASTRAN eight years ago, as the development contract was let, many of our colleagues were skeptical of its projected utility. It would be so large and unwieldly, they said, that we would not be able to work with it, to maintain it, or upgrade it. They have been proven wrong because the program system was well designed and did not "grow like Topsy".

The fact that NASTRAN does well on all the items listed in Figure 4 has made possible its effective use among a large community of engineers. NASA maintains a NASTRAN Systems Management Office at NASA Langley to maintain and upgrade the NASTRAN program; the Navy has a NASTRAN Systems Office at NSRDC which maintains close liaison with the NASA Langley Office and provides many essential supplemental services (e.g., training and consultation) for the Navy and the much larger DOD community. At

---

4. MacNeal, R. H. (Editor), The NASTRAN Theoretical Manual, NASA SP-221(01), Dec 1972.

1. INPUT PROCESSING - INCLUDES DATA GENERATION AND VERIFICATION

2. STRUCTURAL MATRIX ASSEMBLER -

   LOAD VECTOR ASSEMBLER -

   USES FINITE ELEMENT LIBRARY AND MATRIX MANIPULATION PACKAGE.

3. EQUATION SOLVING -

   USES LIBRARY OF EQUATION SOLVERS.

4. OUTPUT PROCESSING - PLOTTED RESULTS ARE ESPECIALLY IMPORTANT.

Figure 3 - Major Parts of a General Finite Element Program

1. SYSTEM DESIGN - MODULARITY - PROVISION FOR MAINTENANCE AND UPGRADING.

2. PROGRAM AVAILABILITY AND ACCESSIBILITY - DOCUMENTATION, CONSULTING AND TRAINING.

3. RANGE OF APPLICABILITY; EASE AND COST OF USE.

4. SOUNDNESS OF MATHEMATICAL METHODS USED; PROVISION FOR ERROR CHECKING.

5. SIZE OF USER GROUP WHICH SHARES EXPERIENCES, MAINTENANCE, AND DEVELOPMENT COSTS.

Figure 4 - Factors Influencing Effective Use of Large Computer Programs

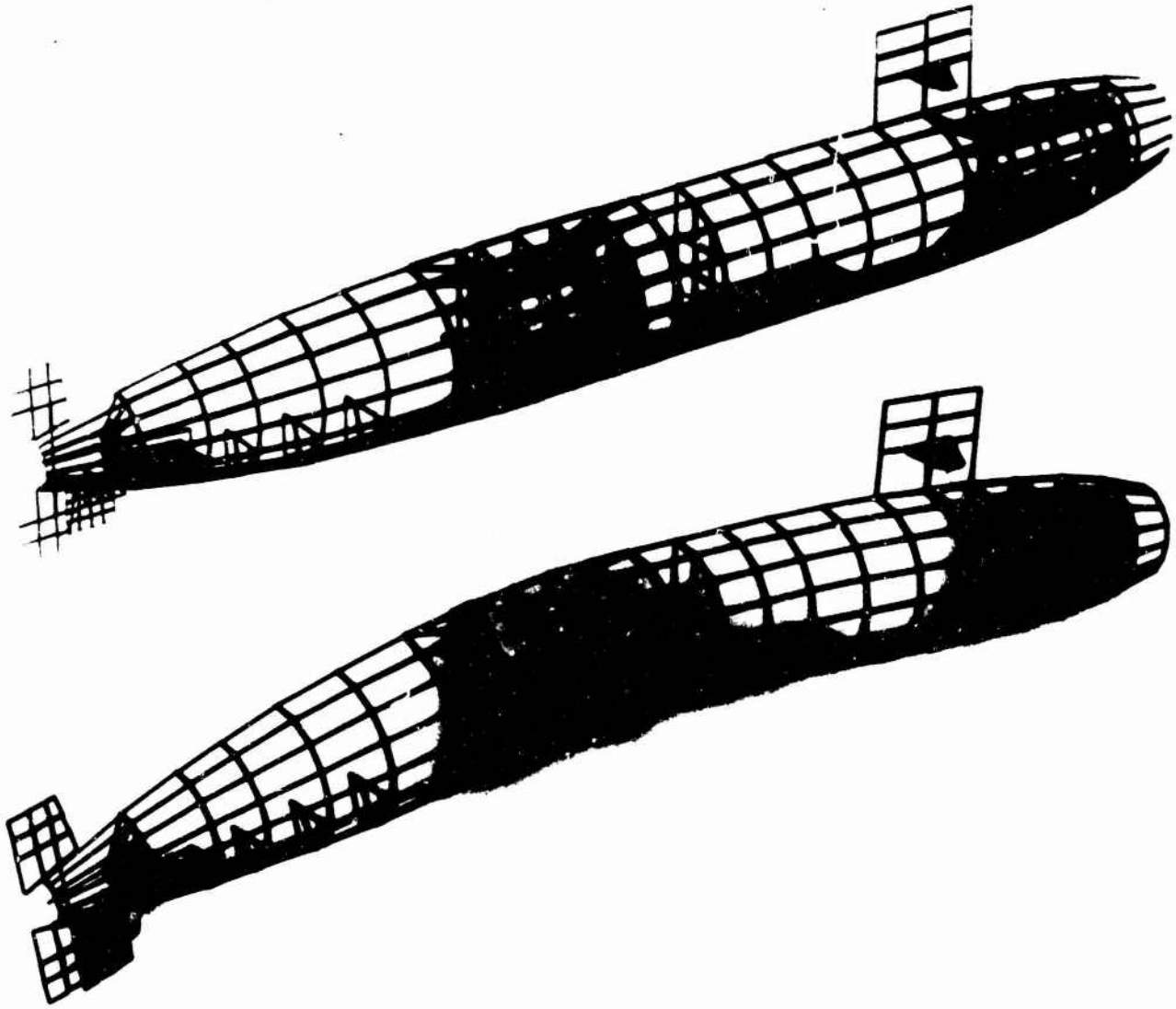
present, the DOD community of NASTRAN users includes nearly every Navy Laboratory as well as many Army and Air Force Laboratories.

Figures 5-7 show a few finite element models that have been analyzed with NASTRAN. Figure 5 shows a submarine model for which some dynamic analyses were run. It gives a picture of the physical model, a NASTRAN plot of the finite element model, and the first flexural mode of vibration calculated and plotted by NASTRAN.

Figures 6 and 7 are from a presentation at our last Navy NASTRAN Colloquium[5] on NASTRAN applications to Army gun components by Frank John of the Benet Weapons Laboratory. Figure 6 shows the modeling with plate elements used for analysis of a muzzle brake. Figure 7 shows the modeling of a breach ring and screw block with NASTRAN solid elements.

A major advantage of working with a widely used program is that one can afford to put much more effort into improving its accessibility to the user and into improving the efficiency of the mathematical and numerical methods used, since the cost of such major improvements are more readily justified when benefits to a large user community are considered. For example, both a general data generation program for NASTRAN[6] and an interactive graphics program[7] (Figure 8) for use in checking NASTRAN input data were developed at NSRDC. The latter permits the finite element model of the structure or any portion of the model to be viewed from any angle. This program has considerably reduced the engineer's time as well as the elapsed time required for data verification for many complex structures.

5. Proceedings of the Fourth Navy-NASTRAN Colloquium, Mar 1973, DDC AD764508.
6. McKee, J. and E. T. Marcus, A General Purpose Data Generator for Finite Element Analysis, Naval Ship Research & Development Center Report #4066, Apr 1973.
7. Kelly, B. M., IDEAL - An Interactive Graphics Aid in the Idealization of a Structural Model, Naval Ship Research & Development Center Report #4014, in preparation.



STRUCTURAL VIBRATIONS
(NASTRAN)

Figure 5 - Submarine Model - Structural Vibrations (NASTRAN)

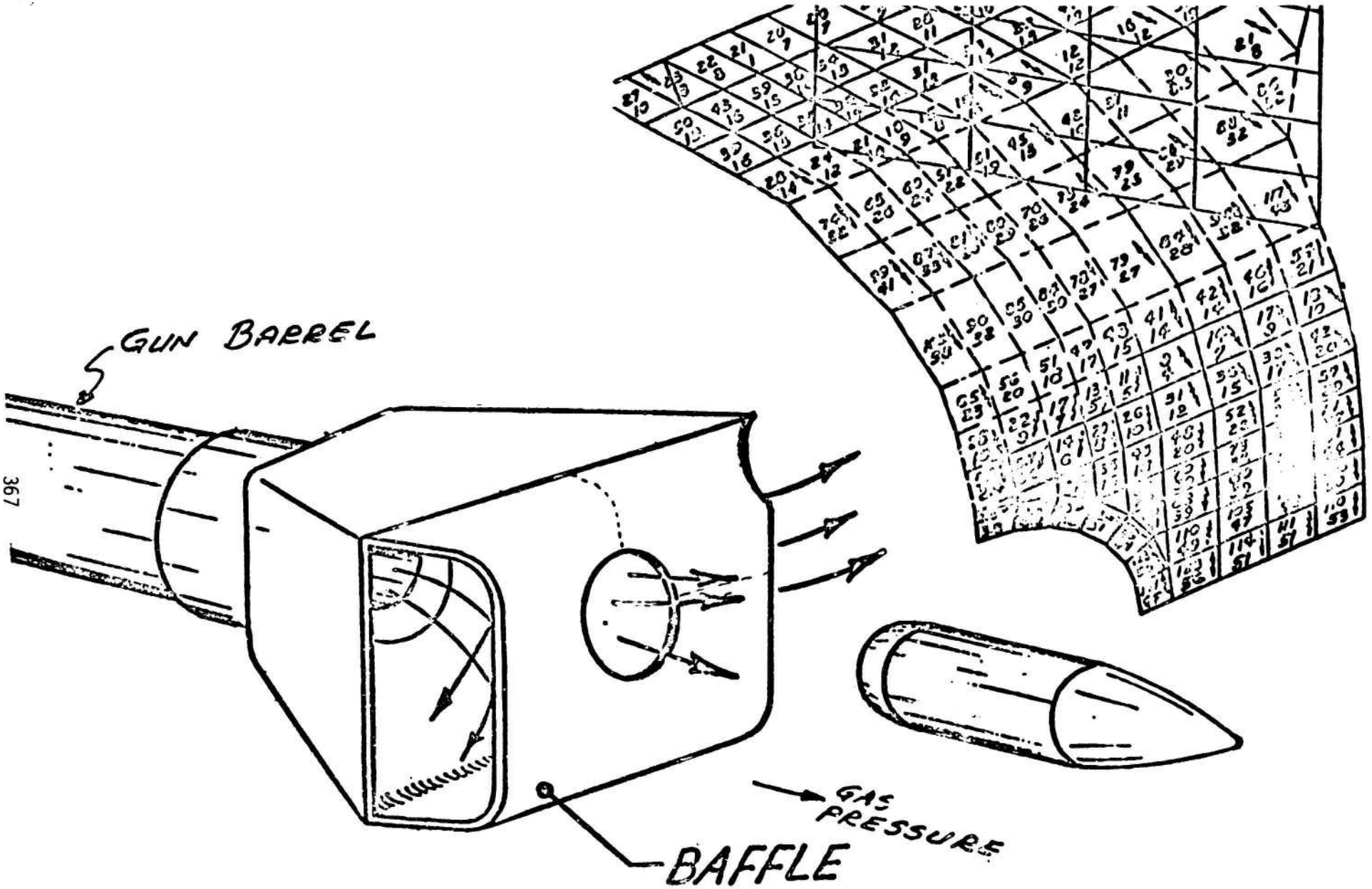
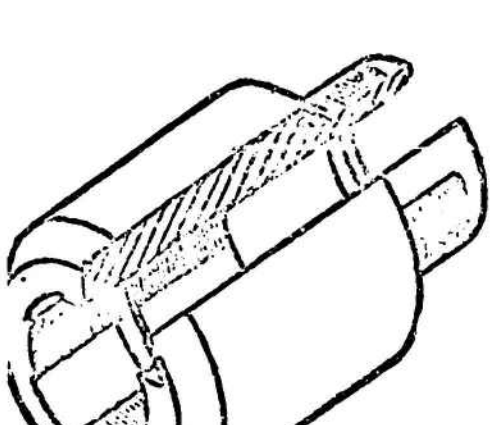
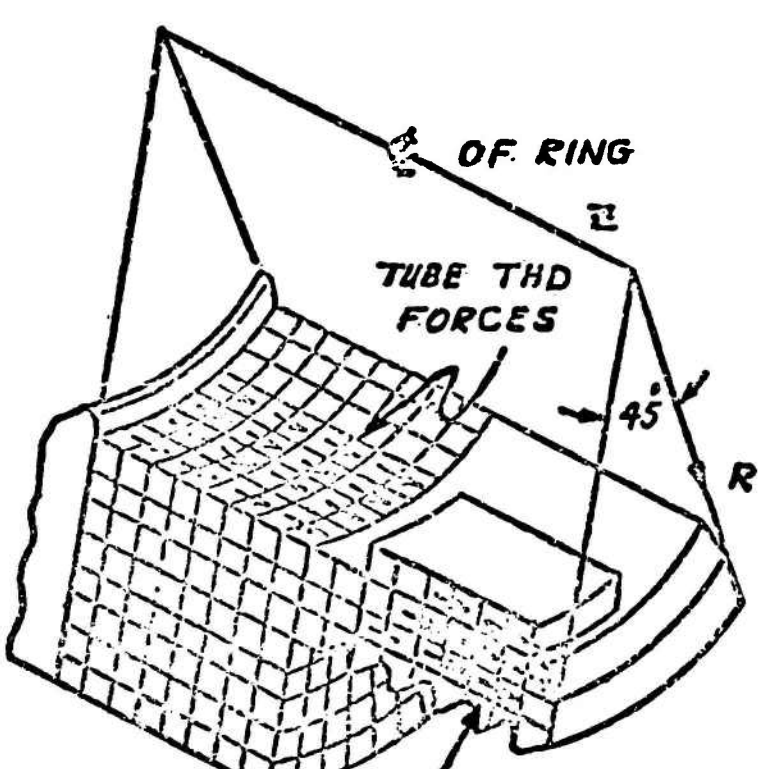


Figure 6 - Modeling a Muzzle Brake with Plate Elements

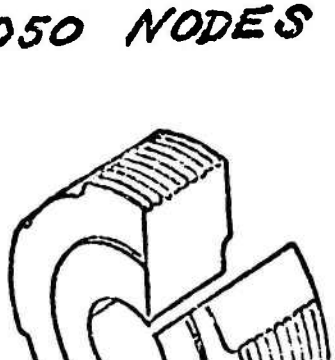
XM199 BREECH RING
NASTRAN 3-D
F.E. ANALYSIS



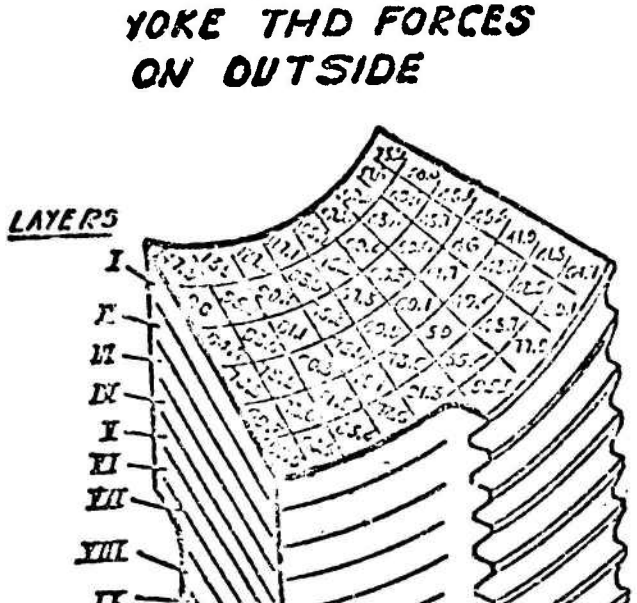


756 ELEMENTS
1050 NODES

YOKE THD FORCES
ON OUTSIDE



Proposed Four-Secter
Screw Block



449 ELEMENTS
660 NODES

Figure 7 - Modeling a Breech and Screw Block with Axisymmetric Solid Elements

Figure 8 - Use of an Interactive Graphics Program

The initial effective design of NASTRAN, the ease of using it, the soundness of the numerical methods used, the maintenance, consultation and training services available, the growing NASTRAN user communities which are sharing and building upon each other's experiences, and the continued upgrading of the system have all contributed to its increasingly effective use. Solutions to many structural problems, which only a few years ago were considered to be research problems, can now be considered routine.

D. EFFICIENT SOLUTION OF LARGE SPARSE SYSTEMS OF EQUATIONS

The last topic of this overview relates to the proper sequencing of both equations and unknowns for efficient solution of the large systems of equations arising when finite element methods are used. One of the major advantages of the use of piecewise polynomial trial functions which characterizes finite element methods is that the equations to be solved have sparse coefficient matrices, i.e., matrices with relatively few non-zero elements. For our problems these matrices are also symmetric and positive definite.

Figure 9 shows the relation of the pattern of zero and non-zero elements (which may be submatrices depending on the number of degrees of freedom per grid point) in the stiffness matrix for our set of equations and the grid point labelling scheme used in the finite element model.

The number of non-zero elements introduced to replace zero elements during a Cholesky factorization is called the fill. Figure 10 shows fill patterns that can occur when the equation systems arising are solved. If the matrix A has the sparsity structure shown and a Cholesky factorization into triangular factors is made, the lower triangular factor is full. All zero elements in the lower triangle of the original matrix become non-zero. If we reorder equations and unknowns with the

$$PAP^T = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \left[\begin{matrix} X & X & X & 0 & 0 & 0 & 0 & 0 \\ X & X & X & X & 0 & 0 & 0 & 0 \\ X & X & X & X & X & 0 & 0 & 0 \\ 0 & X & X & X & X & X & 0 & 0 \\ 0 & 0 & X & X & X & X & X & 0 \\ 0 & 0 & 0 & X & X & X & X & X \\ 0 & 0 & 0 & 0 & X & X & X & X \\ 0 & 0 & 0 & 0 & 0 & X & X & X \end{matrix}\right] \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix} \end{array}$$

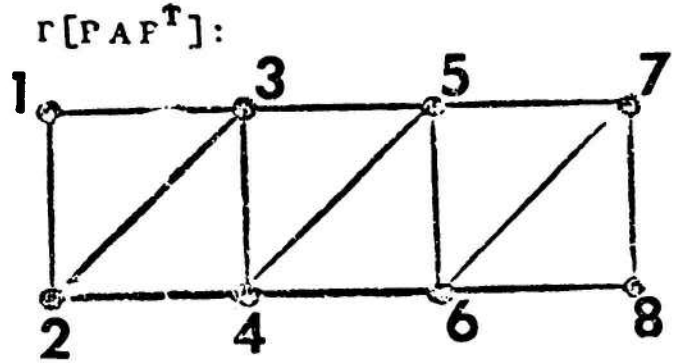


Figure 9 - Pattern of Zero and Non-Zero Elements in a Stiffness Matrix

$$A x = b$$

$$L L^t x = b$$

$$\bar{A} = P A P^t$$

$$\bar{x} = P x$$

$$\bar{b} = P b$$

$$\bar{L} \bar{L}^t \bar{x} = \bar{b}$$

$$A = \begin{bmatrix} \# & \# & \# & \# & \# \\ \# & \# & 0 & 0 & 0 \\ \# & 0 & \# & 0 & 0 \\ \# & 0 & 0 & \# & 0 \\ \# & 0 & 0 & 0 & \# \end{bmatrix} \qquad L = \begin{bmatrix} \# & 0 & 0 & 0 & 0 \\ \# & \# & 0 & 0 & 0 \\ \# & \# & \# & 0 & 0 \\ \# & \# & \# & \# & 0 \\ \# & \# & \# & \# & \# \end{bmatrix}$$

$$\bar{A} = \begin{bmatrix} \# & 0 & 0 & 0 & \# \\ 0 & \# & 0 & 0 & \# \\ 0 & 0 & \# & 0 & \# \\ 0 & 0 & 0 & \# & \# \\ \# & \# & \# & \# & \# \end{bmatrix} \qquad \bar{L} = \begin{bmatrix} \# & 0 & 0 & 0 & 0 \\ 0 & \# & 0 & 0 & 0 \\ 0 & 0 & \# & 0 & 0 \\ 0 & 0 & 0 & \# & 0 \\ \# & \# & \# & \# & \# \end{bmatrix}$$

Figure 10 - Typical Fill Patterns in Matrix Factorization

appropriate permutations, in this particular case, no non-zero elements need be introduced. For the matrix factorization shown in Figure 10, there is a fill of 6 in the first case and 0 in the second. Such considerations can have significant repercussions on the amounts of storage and calculation required when such systems, which can involve hundreds and thousands of unknowns, are solved directly.

Several observations should be made. First, it can be shown that all fill will occur in a band about the main diagonal which extends to the element farthest from the diagonal. In this case, since the upper right hand corner element is non-zero, this is no help. However, for the matrix shown in Figure 9 it does help, since the matrix has a "banded" structure. In that case all elements within the band are non-zero, so that we can predict a fill of zero in this case. Many equation solvers have been written to take advantage of banded matrix structure. Matrix storage schemes used by such solvers are simple and involve little overhead[8,9,10].

It can also be shown that all fill occurs between the first non-zero element of any column and the diagonal for any upper factor or, equivalently, between the first non-zero element of any row and the diagonal for the lower factor. One might provide storage for just these elements and set up an equation solver to take advantage of the reduced computation needed. Such a program requires more overhead, but clearly such a scheme will handle sparsity structures such as that in Figure 10 much more effectively. Such schemes are often called

---

8. Cuthill, E. H., "Several strategies for reducing the bandwidth of matrices", in Rose, Willoughby, eds., Sparse Matrices and Their Applications, Plenum Press, New York, 1972.

9. Gignac, D., Comparative Study of Several Core Storage Schemes for Large Sparse Positive Definite Matrices with Reference to the Cholesky Algorithm, Naval Ship Research and Developmt Cntr. Rept #4017, Nov. 1972.

10. Gignac, D., SOLVEDG, An Out-of-Core System Solver for Large Order Positive Definite Systems of Linear Equations, Naval Ship Research and Development Center Report #4235, Aug. 1973.

frontal or profile schemes,[8,11,12]

We can also work toward minimizing fill directly. The overhead here will be greater still, but the benefits from such an approach[8,13,14] can be significant.

Figure 11 shows an actual structure with the corresponding pattern of non-zero elements generated as a result of the original grid point labeling. The pattern on the right shows the results obtained when the grid point labels were resequenced using a very fast resequencing scheme developed at the Center. This scheme was incorporated into a computer program called BANDIT[15]. Note that the pattern of non-zero elements clusters much more closely about the diagonal, producing a narrower bandwidth after resequencing. An equation solver such as that in NASTRAN which takes advantage of narrowed bandwidth can generate a solution much faster, in this case in roughly 10% of the computing time which would be required for solution of the system of equations without resequencing.

---

11. Gignac, D., CSKYDG: An Out-of-Core Cholesky Algorithm Equation Solver for Large Positive Definite Systems of Linear Equations, Naval Ship Research and Development Center Report #4377, Feb. 1974.

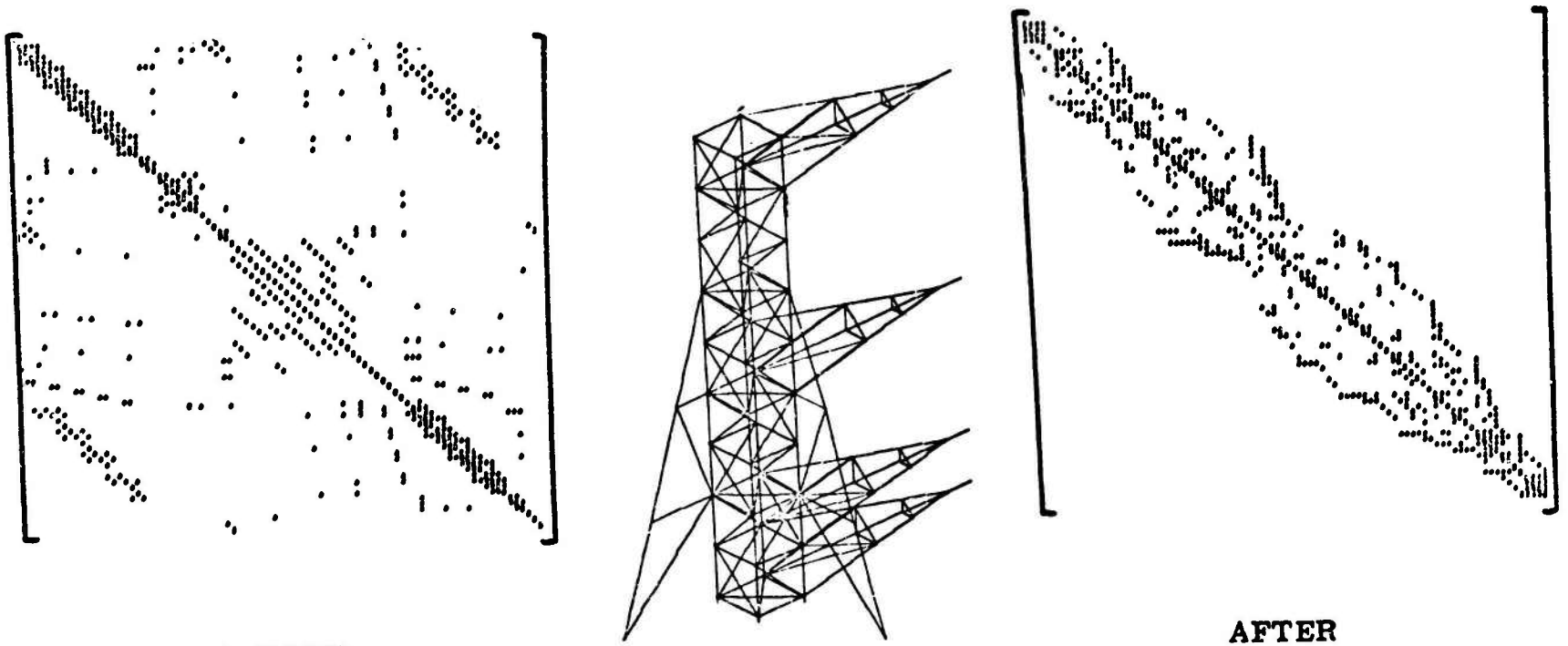
12. George, J. A., Computer Implementation of the Finite Element Method, Ph.D. Thesis, Computer Sci. Department, Stanford Univ., Stanford, California, 1971.

13. Birkhoff, G., and A. George, "Elimination by nested dissection", in Complexity of Sequential and Parallel Numerical Algorithms, Academic Press, N. Y., 1973.

14. Rheinboldt, W. C., and C. K. Meztenyi, Arc Graphs and Their Possible Application to Sparse Matrix Problems, Technical Report TR-238, University of Maryland Computer Science Center, College Park, Maryland, Apr. 1973.

15. Everstine, G. C., The BANDIT Computer Program for the Reduction of Matrix Bandwidth for NASTRAN, Naval Ship Research and Development Center Report #3827, Mar. 1972.

BANDIT EXAMPLE

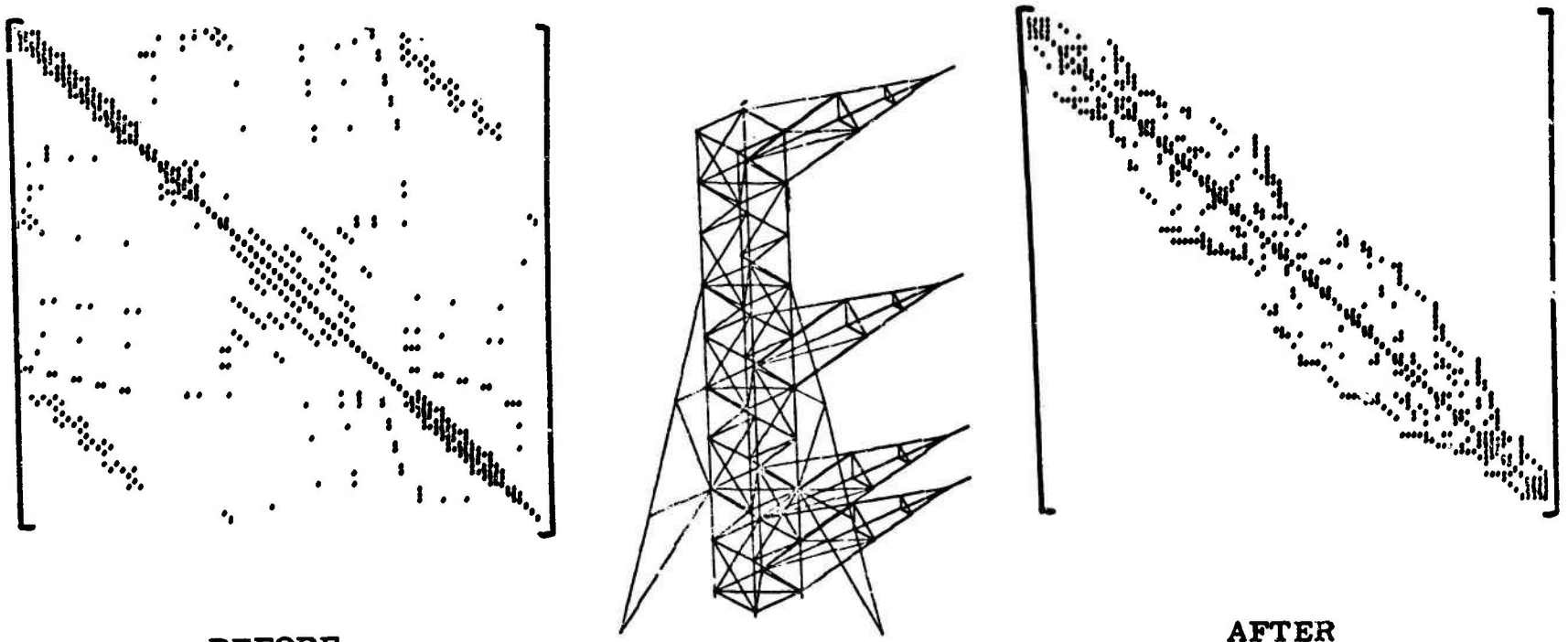


BEFORE
(MATRIX BANDWIDTH = 63)

AFTER
(MATRIX BANDWIDTH = 17)

Figure 11 - Stiffness Matrix before and after Grid Point Resequencing by BANDIT

BANDIT EXAMPLE



BEFORE
(MATRIX BANDWIDTH = 63)

AFTER
(MATRIX BANDWIDTH = 17)

Figure 11 - Stiffness Matrix before and after Grid Point Resequencing by BANDIT

E. REFERENCES

1. Zienkiewicz, O.C., The Finite Element Method in Engineering Science, 2nd Ed., McGraw-Hill, New York, 1971.
2. Strang, G., and G. J. Fix, An Analysis of the Finite Element Method, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.
3. Schultz, M. H., Spline Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.
4. MacNeal, R. H. (Editor), The NASTRAN Theoretical Manual, NASA SP-221(01), Dec 1972.
5. Proceedings of the Fourth Navy-NASTRAN Colloquium, DDC AD 764508, Mar 1973.
6. McKee, J., and E. T. Marcus, A General Purpose Data Generator for Finite Element Analysis, Naval Ship Research and Development Center Report #4066, Apr 1973.
7. Kelly, B. M., IDEAL - An Interactive Graphics Aid in the Idealization of a Structural Model, Naval Ship Research & Development Center Report #4014, in preparation.
8. Cuthill, E. H., "Several strategies for reducing the bandwidth of matrices", in Rose, Willoughby, eds., Sparse Matrices and Their Applications, Plenum Press, New York, 1972.
9. Gignac, D., Comparative Study of Several Core Storage Schemes for Large Sparse Positive Definite Matrices with Reference to the Cholesky Algorithm, Naval Ship Research & Development Center Report # 4017, Nov. 1972.
10. Gignac, D., SOLVEDG, An Out-of-Core System Solver for Large Order Positive Definite Systems of Linear Equations, Naval Ship Research and Development Center Report #4235, Aug 1973.
11. Gignac, D., CSKYDG: An Out-of-Core Algorithm Equation Solver for Large Positive Definite Systems of Linear Equations, Naval Ship Research & Development Center Report #4377, Feb 1974.

12. George, J. A., Computer Implementation of the Finite Element Method, Ph.D. Thesis, Computer Sci. Department, Stanford Univ., Stanford, California, 1971.

13. Birkhoff, G., and A. George, "Elimination by nested dissection", in Complexity of Sequential and Parallel Numerical Algorithms, Academic Press, N. Y., 1973.

14. Rheinboldt, W. C., and C. K. Meztenyi, Arc Graphs and Their Possible Application to Sparse Matrix Problems, Technical Report TR-238, University of Maryland Computer Science Center, College Park, Maryland, Apr 1973.

15. Everstine, G.C., The BANDIT Computer Program for the Reduction of Matrix Bandwidth for NASTRAN, Naval Ship Research & Development Center Report #3827, Mar 1972.

APPLICATION OF NONLINEAR ANALYSIS (PLASTIC) TO NASTRAN (NASA STRUCTURAL ANALYSIS) USING RING ELEMENTS INCLUDING ASPECT RATIO EFFECTS

Diana L. Frederick
Munitions Development & Engineering Directorate
U.S. Army
Frankford Arsenal
Philadelphia, Penna. 19137

ABSTRACT. NASTRAN is a general-purpose digital computer developed by NASA for application to almost any type of linear and some nonlinear structures that can be represented by combinations of elements contained in the NASTRAN library, such as beams, rods, shells, etc.

A wide range of analysis capability has been built into NASTRAN. The capability for solving nonlinear problems is limited because it does not include ring elements. In order to model ammunition problems, this is a requirement.

A useful and practical exercise of nonlinear analysis using ring elements in the field of ammunition structural analysis is the interaction between the cartridge case and gun barrel chamber. In particular, the model enables the effects of changing tolerances, material mechanic properties, and geometrical variations on the distribution of stresses to be conveniently estimated.

Three studies were performed to test the applicability of NASTRAN. Study I modified NASTRAN to perform a nonlinear analysis (plastic) and include RING elements. Study II demonstrated that large Aspect Ratios in RING elements can be safely ignored. Study III showed that varying element sizes within a model does not effect deformation of stresses.

1. INTRODUCTION. The NASA structural analysis (NASTRAN) digital computer program, a general purpose digital computer developed by the National Aeronautics and Space Adminstration, is designed to analyze the behavior of elastic structures under a range of loading conditions, using a finite-element displacement method approach. The program is applicable to almost any type of linear and some nonlinear structures that can be represented by combinations of elements contained in the NASTRAN library, such as beams, rods, shear and twist panels, triangular and quadrilateral plates, conical and toroidal shells, solids of revolution, scalar elements, general elements, and constraint elements.

A wide range of analysis capability has been built into NASTRAN, including static response to concentrated and distributed loads, to thermal expansion, and to enforced deformation; dynamic response to transient

loads, to steady-state sinusoidal loads, and to random excitations; determination of real and complex eigenvalues for use in vibration analysis, dynamic stability analysis, and elastic stability analysis. In addition, there is a limited capability for solving nonlinear problems, including piece-wise linear analysis of nonlinear static response and transient analysis of nonline dynamic response.

The piece-wise linear analysis option of NASTRAN is used to solve problems on material plasticity. The load is applied in increments such that the stiffness properties can be assumed to be constant over each increment. The stiffness matrix for each increment is dependent on the current state of stress in the structural elements. The increments in displacements and stresses are accumulated to produce the final nonlinear results.

The nonlinearity of a structural element is defined by the material characteristics of the material elements. Any isotropic material may be made nonlinear by including a stress-strain table, defining its extension test characteristics. The stress-strain table must define a nondecreasing sequence of both stresses and strains. Because the stiffness matrix for the first load increment uses the elastic material coefficients, the initial slope should correspond to the defined modulus of elasticity, E. Linear elements and materials may be used in any combination with the nonlinear elements. Linear elements are used in a more efficient manner than the nonlinear elements since there are not extrapolations. The nonlinear effects depend on the element type. The elements which utilize the plastic material properties are ROD, TUBE, BAR and PLATE elements.

2. DISCUSSION. A useful and practical exercise of nonlinear analysis in the field of ammunition structural analysis, using RING elements, is the interaction between the cartridge case and gun barrel chamber. In particular, the model enables the effects of changing dimensional tolerances, material mechanical properties, and geometrical variations on the distribution of stresses to be conveniently estimated.

A characteristic problem is a case neck separation (CNS) malfunction explanation. Specifically of interest is the rupture or separation which occurs at the neck-shoulder section. This is not to be confused with separations that occur at the crimp between the case and projectile. This section is potentially a future candidate for an exercise example.

As background, all reported CNS have occurred in M61 (6 barrel) or M197 (3 barrel) type Gattling guns. No CNS have been reported from firings in the M39 type revolver gun; no CNS have been reported from case and cartridge acceptance firing tests at either Lake City Army Ammunition Plant (LCAAP) or at Aberdeen Proving Ground (PAG); no CNS have been reported with the firing of M56 (HEI) ammunition. All CNS have occurred with M55 (TP) and M220 (TPT) cartridges, and all have occurred with cases manufactured by a particular contractor.

CNS present serious problems in double ended linkless feed systems, i.e., system in which the fired cases and released rounds are returned to the storage drum. Round control of cases with partially or completely

separated necks is lost by the hand-off sprockets, resulting in a system jam. In systems where the fired cases are dumped overboard, CNS present no problem unless the separated case neck remains in the gun barrel chamber, which can result in a jam when a subsequent round is fed into the same barrel.

CNS were initially reported in early 1972 in A7D aircraft, which have an internally mounted M61A1 gun with a double ended linkless feed system. Results of the malfunction investigation indicated that the CNS were caused by the neck-shoulder blend radius in the gun barrel chamber being out of drawing tolerance on the low side. The barrel drawing calls for a 0.250 + 0.125 in. blend radius; barrels from guns where CNS occurred measured from 0.040 + 0.125 in. However, in subsequent malfunction investigations, CNS occurred in barrels within specification neck-shoulder blend radius. Results of numerous firing tests indicate that a sharp or under specification blend radius can increase the frequency of CNS but is not the cause. The NASTRAN model exercise is designed to include both material properties and geometry variations.

When a round is fired, the powder pressure builds up and the sidewall expands elastically to its yield point and then completes its expansion plastically. Although the sidewall may or may not enter the plastic range before taking up the initial clearance between the case and the chamber, it will be completely plastic when the pressure reaches its maximum value. At this instant of maximum pressure, both the case outside diameter and chamber inside diameter will have expanded together to a common maximum value. Here the cartridge case sidewall will be acted upon on the inside by the internal pressure and on the outside by the chamber-cartridge case interface friction and pressure. The chamber wall will be acted upon by equal and opposite friction and pressure. Knowing the radial loads on the cartridge case at this instant of maximum pressure, the associated state of stress in the sidewall can be determined for various assumed values of axial (longitudinal) stress in the sidewall. This is done by applying either the Von Mises or the Tresca law of yielding, together with its associated flow rule.

In the problem of expansion of the wall of cartridge case and barrel chamber by the pressure of propellant gases and the stress analysis of the structure, it is desired that the axi-symmetric solid of revolution RING element be utilized. This element offers both simplicity and accuracy over other elements. An explanation of this structure element is given in Section 3, and a demonstration problem is given in Reference 3. Since the piece-wise linear analysis has not been developed for this element, a study was initiated to perform the piece-wise linear analysis manually. A summary flow diagram is given in Figure 1.

The various steps, following similar operations used in Rigid Format 6 (see References 2 and 3), are given numbers corresponding to the explanation below.

1. The normal statics analysis is used to generate the grid point and element. The stiffness matrix is generated in the normal manner, using the modulus of elasticity given with the materials.

2. The linear elements are used to generate a lower stiffness matrix, (K) (see Reference 2). This matrix will not change with loading changes.

3. The load vector for the whole structure, (F), is generated by the normal methods (see Reference 2). The constrained points are also identified in this stage.

4. The incremental displacements are generated using the current stiffness matrix and the current load vector increment. The dependent displacements are recovered in the normal manner and merged to produce the increments for all degrees of freedom, (ΔU_i) (see Reference 2). The increments are added to the previous vectors to produce the current vectors.

$$\{U_i\} = \{U_{i-1}\} + \{\Delta U_i\}$$

5. Incremental element stresses are calculated. The increments are added to the previous vectors to produce the current vectors.

6. The Tresca yield criterion and its associated flow rule are used.

7. Based on the calculated stresses, the modulus of elasticity for the nonlinear elements are calculated from the stress-strain curves and replaces the original modulus. The new stiffness matrix is generated.

8. For linear elements, keep everything the same.

9. Select next load increment and rerun the problem.

Three sample cases provided data for verifying the validity of this piece-wise linear manual approach. The first case consisted of a static analysis of a five-element truss with an applied force of 100,000 pounds. Each element has its own elastic-plastic material property. Comparisons are made with the piece-wise linear analysis, using Rigid Format 6. The second and third cases dealt with the elastic-plastic analysis of an open ended thick wall and an open ended thin wall cylinder, exposed to a high internal pressure. Comparisons are made with known mathematical solutions. The effect of different aspect ratio, using the axi-symmetric solid of revolution RING element was also investigated. The accuracies of the results are again checked by comparing with known mathematical solutions.

3. STRUCTURE ELEMENT - RING. The triangular RING element is defined with a CTRIARG card. No property card is used for this element. The material property reference is given on the connection card. The integers 1, 2, and 3 on Figure 2 refer to the order of the connected grid points on the CTRIARG card. This order must be counterclockwise around the element. The grid points must lie in the r-z plane of the basic cylindrical coordinate system, and they must lie to the right of the axis symmetry.

The radial and axial forces at each connected grid point are output on request. The positive directions for these forces are shown in Figure 2. These are apparent element forces, and they include any equivalent thermal loads. The stresses at the centroid of an element are output on request. The available quantities are the normal stresses in the radial, circumferential, and axial directions and the shear stress on the radial face in the axial direction. Positive stresses are in the positive direction on the positive face.

The coordinate system for the trapezoidal ring element is shown in Figure 2. This element is similar to the triangular RING element. This element has the additional restriction that the element numbering must begin at the lower left-hand corner of the element. Also, the parallel faces of the trapezoid must be perpendicular to the axis of symmetry. This element can be used in the limiting case where the r coordinates associated with grid points 1 and 4 are zero. In this special case, the element is referred to as a core element.

The trapezoidal RING element is defined with a CTRIARG card in a manner similar to that for a triangular element. The forces at the four connected grid points are provided on request in a manner similar to that for a triangular element. In addition to providing the stresses at the centroid of the trapezoid, similar stresses are provided at the four connected grid points.

4. ALTERED RIGID FORMAT AND SAMPLE RUN. In order to perform the piecewise linear analysis manually, the rigid format must be altered to enable the user to store data output from one run to use as data input for the next run. To use files rather than tapes for data storage, the following card must be inserted before the Executive Control Deck, (see Reference 1).

```
NASTRAN - SYSTEM (45)_ = _ 384 ___$
```

For Run 1 the following cards are inserted in Executive Control Deck in order to change the rigid format.

```
ALTER - 110

OUTPUT1   ___ ___,,,,//C,N, 1/C,N, Ø/C,N,USERPLA__ __$

End Alter
```

For Run 2

```
ALTER_ 110

INPUT1 -/UGPREV,,,,/C,N,-1/C,N,1/C,N,USERPLA_$
```

```
ADD__ __ UGPREV, UGV/UGVV__ __$
OUTPUT1, ___ __,,,,//C,N,-1/C,N,Ø/C,N,USERPLA__ __$
OUTPUT1 ___ ___ UGVV,,,,//C,N,Ø/,C,N,Ø/C,N, USERPLA__ __$

ALTER__121

SDR2_CASE CC, CSTM, MPT, DIT, EOEXIN, SIL, GPTT, EDT, BGPDT, PGG,ØG, UGVV,
EST,/

__ __ OPG2, OØG2, OUGV2, OES2, OEF2,/C,N,STATICS _$

OFP__OUGV2, OPG2, OØG2, OEF2, OES2,//v,n, CARDNO/V,Y, OPTION_$

END ALTER
```

The following control cards are needed for Scope 3.4.1 on CDC 6500.

Run 1

```
Rewind, INPT

NASTRAN.  ATTACH

Rewind, INPT

Catalog, INPT, Cylinder def, cy=1, ID = Frederick
```

Run 2

```
Attach, INP1, cylinder def, cy=1

Rewind, INPT, INP1.

NASTRAN.  Attach

Rewind, INPT

Catalog, INPT, Cylinder def, cy=2, ID = Frederick
```

With Run 1, the user has cataloged the output in order to be used for input to Run 2. In Run 2 it can be observed that the "ALTER 110" uses the input from Run 1 USERPLA and creates output USERPLB. There are also "ALTER 121" cards. These cards allow the incremental stresses, forces, and displacement to be printed out. For Run 3, only three cards will be changed: change USERPLA to USERPLB on INPUTTI and change USERPLB to USERPLC on both OUTPUT 1 Cards.

5. ASPECT RATIO ANALYSIS. During the course of the investigation it was realized that the aspect ratio would be very large for the finite elements to be used in the NASTRAN model of the cartridge case. The aspect ratio is defined as the ratio of the element length to its height. From past experience in using NASTRAN and from consultation with NASTRAN experts, it was more or less understood that the aspect ratio for PLATE elements should not exceed three, and this is most likely true for the RING elements. However, since RING elements are seldom used, no basic study on the applicability of the RING elements as a function of aspect ratio has been made; therefore, a study was required to determine if the problem of aspect ratio in RING elements could be ignored.

The finite elements used in synthesizing the NASTRAN models of the thick (steel) and thin (brass) wall cylinders are shown in Figures 3 and 4. The tube was assumed to be free at both ends, with internal pressure applied. The materials for the thick and thin wall models were steel and brass, with the same Poisson's ratio of 0.3. Only elastic analysis was considered.

A list of all the cases investigated for various aspect ratios and the total number of elements is given in Figure 5. The results from the NASTRAN analysis for displacements and radial and circumferential stresses are compared with the theoretical results, using the classical equations for the solution of elastic analysis in thick and thin walled cylinders. (A)

From the results, the problem of aspect ratio that existed in the PLATE elements can be safely ignored with respect to RING elements.

6. NASTRAN PIECE-WISE LINEAR ANALYSIS. Case I. Open Ended Thick Wall Cylinder - The finite elements used in synthesizing the NASTRAN model of the thick wall cylinder is shown in Figure 3. The tube was assumed to be free at both ends, with internal pressure applied. Rigid Format 1 and RING elements were used. The overall model had 85 RINGS (or grid circles) and 64 elements, yielding a total of 151 degrees of freedom. The material for the model is steel, with a Poisson's ratio of 0.3. A bilinear stress-strain curve was selected for the elastic-plastic material property.

A total of 17 runs was made manually for the 16-layer tube. This is the minimum number of runs since each load increment was precalculated to make the material in the subsequent layer plastic. In this problem, elements along the same layer were assumed to behave the same. Although this isn't necessarily true in actual problems, it can still be handled by checking each element and making more runs.

The incremental displacements and stresses were cataloged and filed after each run. These were then added to the previous results to obtain the total displacements and stresses. After each run the stresses were tested with the Tresca yield condition. The elastic material properties of those elements that satified the yield criterion are changed into plastic material properties .

Case II Open Ended Thin Wall Cylinder - In order to gain additional confidence on the manual piece-wise linear approach, a thin wall cylinder was considered since the wall of a cartridge case is very thin with respect to its diameter. The finite elements used in synthesizing the NASTRAN model of the thin wall cylinder are shown in Figure 4. The shell was assumed to be free at both ends, with internal pressure applied. Rigid Format 1 and RING elements were used. The overall model has 25 RINGS (or grid circles) and 16 elements, yielding a total of 45 degrees of freedom. The material selected for this model is brass, with a Poisson's ratio of 0.3 in the elastic range and 0.45 in the plastic range. A bilinear stress-strain curve was selected for the elastic-plastic material property. Only a total of 5 runs was needed since there were only 4 layers through the thickness of the wall. The same procedures were followed as in the thick wall cylinder case.

The significant results from the NASTRAN analysis are presented in Figures 6, 7 and 8. The theoretical elastic-plastic analysis in the thick wall cylinder is used, together with a thin shell analysis, for comparison with the NASTRAN results. Figure 6 shows the comparison of the radial displacement. Figure 7 and 8 show the comparison of the radial and circumferential stresses, respectively, for an internally applied pressure load of 22,069 psi. Reasonable agreements were obtained. This could definitely be improved by increasing the number of elements.

7. ELEMENT SIZE STUDY. In modeling the cartridge case, the elements around the neck are small in size. If all the elements are kept the same size, the number of elements in the cartridge case model is very large. In turn, the run time for the job is long and, thus, expensive. The number of elements is reduced by increasing the aspect ratio of the elements other than those in the cartridge case neck.

The finite elements used in synthesizing the NASTRAN models of the thin cylinders are shown in Figure 9. The tube was assumed to be free at both ends, with internal pressure applied. The material for the thin wall model was brass, with a Poisson's ratio of 0.3. Only elastic analysis was considered.

The results for displacement, and radial and circumferential stresses from the NASTRAN analysis are compared with the theoretical results, using the classic equations for the solution of elastic analysis in cylindrical shells. Similarly the results show that we can decrease the number of elements in our model by increasing the element size in the areas other than those in the cartridge case neck area.

8. SPECIAL APPLICATIONS. Based on the results obtained using the RING element, the manual piece-wise linear analysis appears to give accurate results. We can now proceed to investigate the design of the cartridge case neck and barrel chamber interface section of a high pressure ballistice system. The model can be observed in Figure 10.

In addition to being able to model the exact configuration of the cartridge case and barrel chamber, we are able to include, in this manual piece-wise linear analysis, varying material properties along the wall of the case and chamber. A step by step procedure is described below as the powder pressure builds up and the sidewall expands when a round is fired. (Also shown in Figure 11)

1. Loads are applied incrementally.

2. The sidewall expands elastically to its yield condition. This is done by applying the Von Mises or the Tresca law of yielding, together with its associated flow rule.

3. Every element is being checked after each run to insure that the yield condition has not been exceed.

4. The material properties of those elements which exceeded the yield condition are to be changed appropriately according to the stress-strain curve.

5. This continues until the entire sidewall of the cartridge case becomes plastic. Although the sidewall may or may not enter the plastic range before taking up the initial clearance between the case and the chamber, it will be completely plastic when the pressure reaches its maximum value.

6. The displacement of the outer wall of the cartridge case will be observed very closely for every element. When the value of the displacement of any point reaches the value of the clearance between the case and the chamber, the points will be connected. This will continue until all points on the case are connected to the chamber.

7. From here on the case outside diameter and chamber inside diameter will expand together to a common maximum value at the instant of maximum pressure. Here the cartridge case sidewall will be acted upon on the inside by the internal pressure and on the outside by the chamber-cartridge case interface friction and pressure.

8. A failure criterion will have been set up such that each element will be tested after each run.

9. CONCLUSIONS.

1. Elastic-plastic problems with static loading can be solved using the manual piece-wise linear analysis.

2. The manual piece-wise linear analysis is more accurate than the automated piece-wise linear analysis, Rigid Format 6.

3. RING elements, which are convenient to use for axi-symmetric bodies, appear to give accurate results.

4. With respect to the problems of aspect ratio that existed in the PLATE elements, this can be safely ignored with respect to RING elements.

10. RECOMMENDATIONS.

1. Apply the described procedure to a specific problem as a means for evaluating potential for broad application to ammunition problems.

2. Provide subroutine for eliminating manual computations.

11. REFERENCES.

1. The NASTRAN User's Manual, NASA SP-222, Section 3, September 1970.

2. The NASTRAN Theoretical Manual, NASA SP-221, Section 3, September 1970.

3. NASTRAN Demonstration Problem Manual, NASA SP-224, pp6. 1-1 through 6, 1-12, September 1970.

4. Brophy, J. M., Computer Aided Cartridge Case Design Using Finite Element Stress Analysis: The Automation of Finite Element Configuration, Frankford Arsenal Report R-2054, September 1972

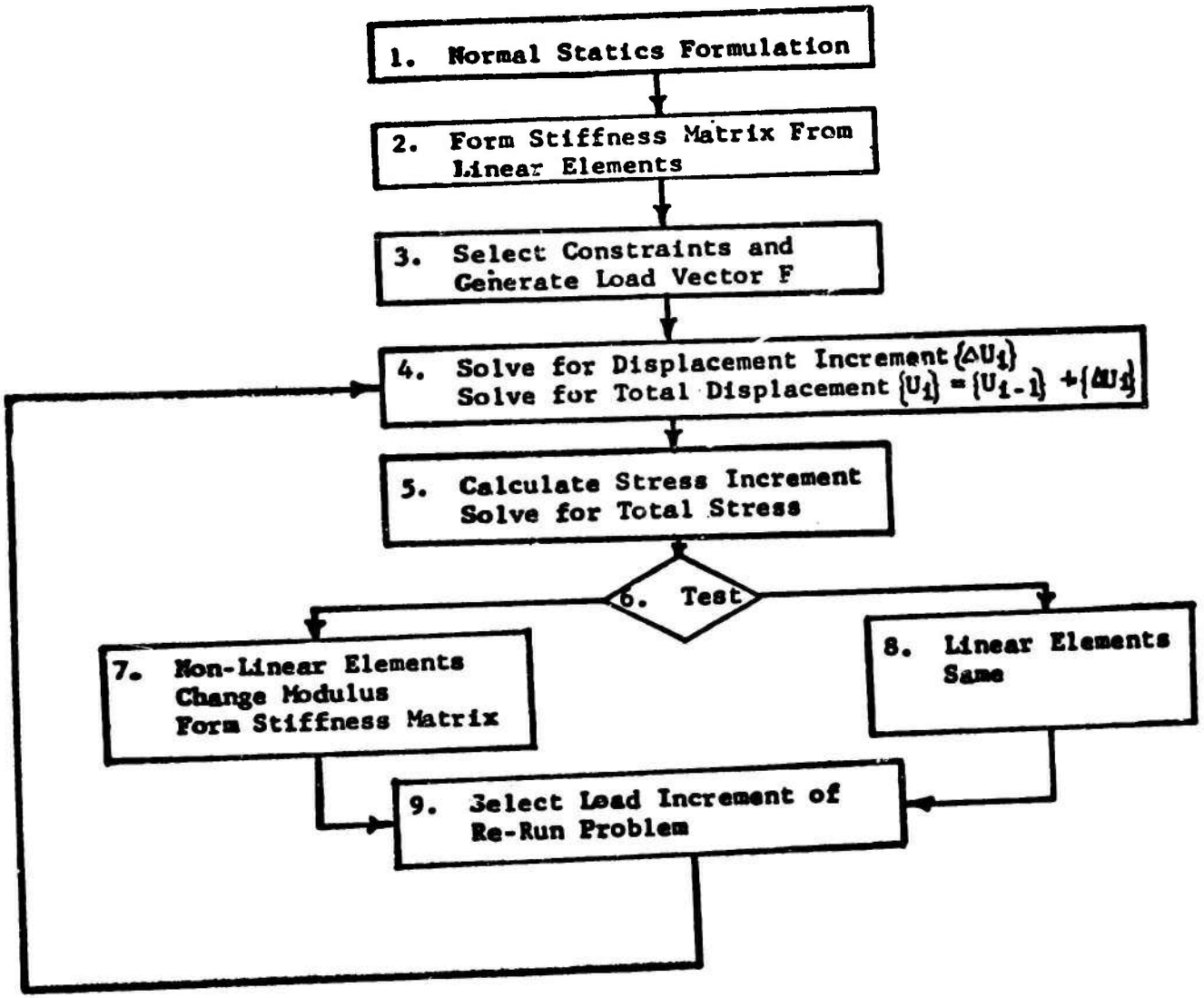


Figure 1. Manual Piece-wise Linear Flow Diagram

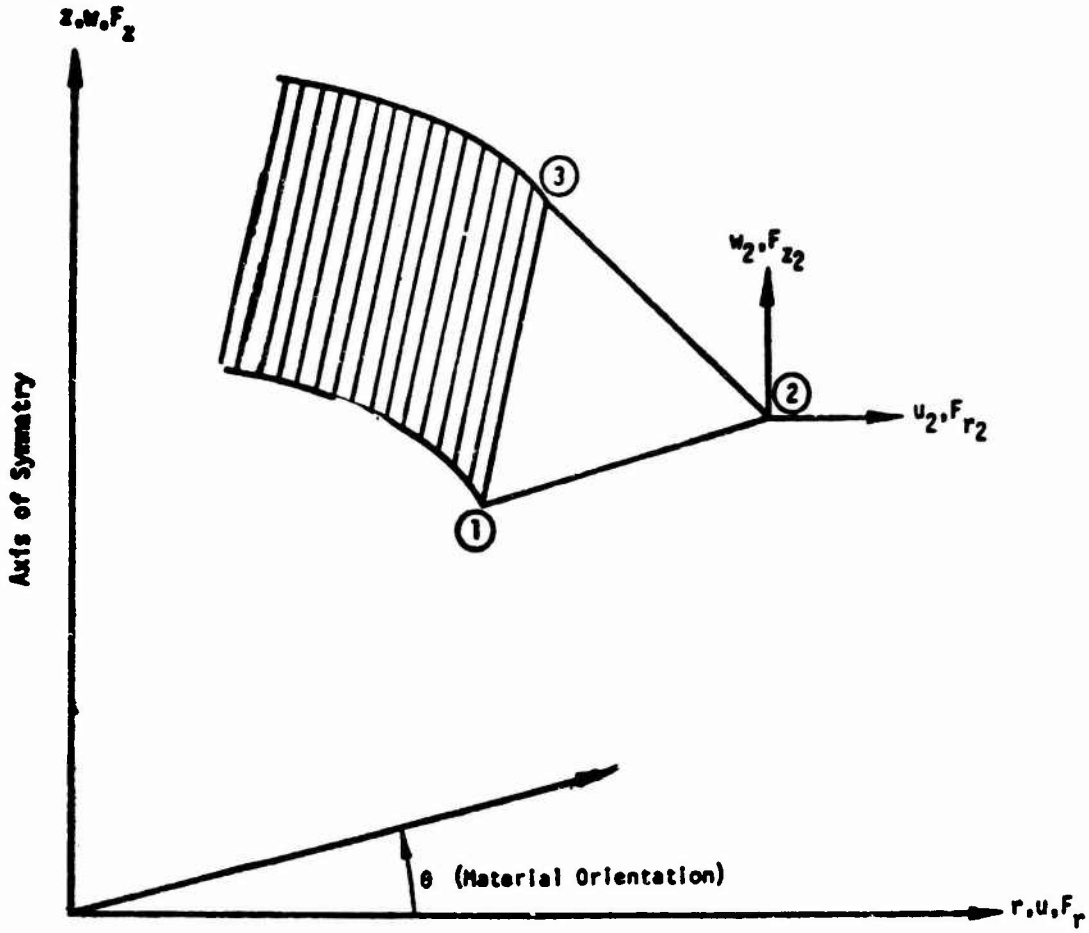


Figure 2-a. **Triangular RING Element Coordinate System**

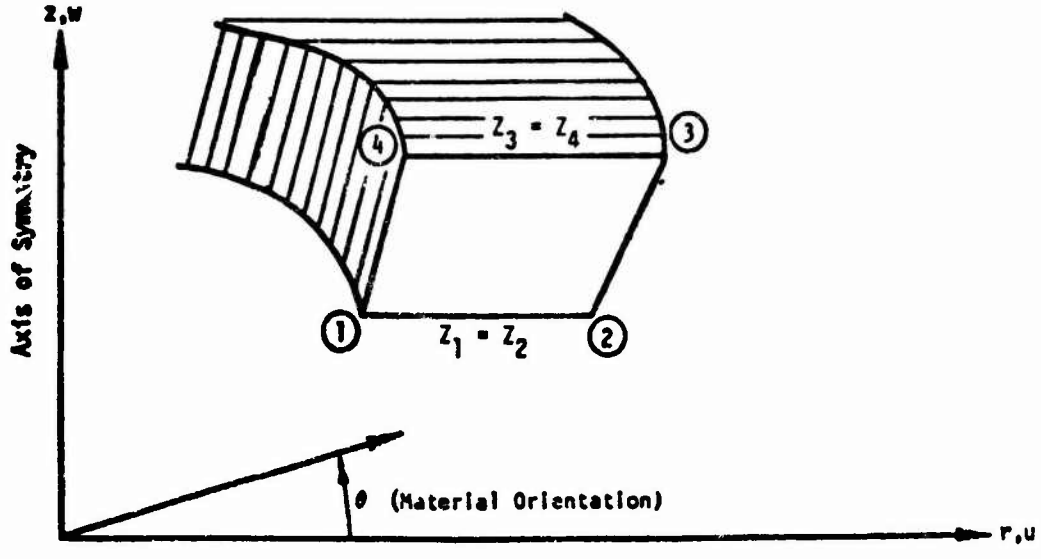


Figure 2-b **Trapezoidal RING Element Coordinate System**

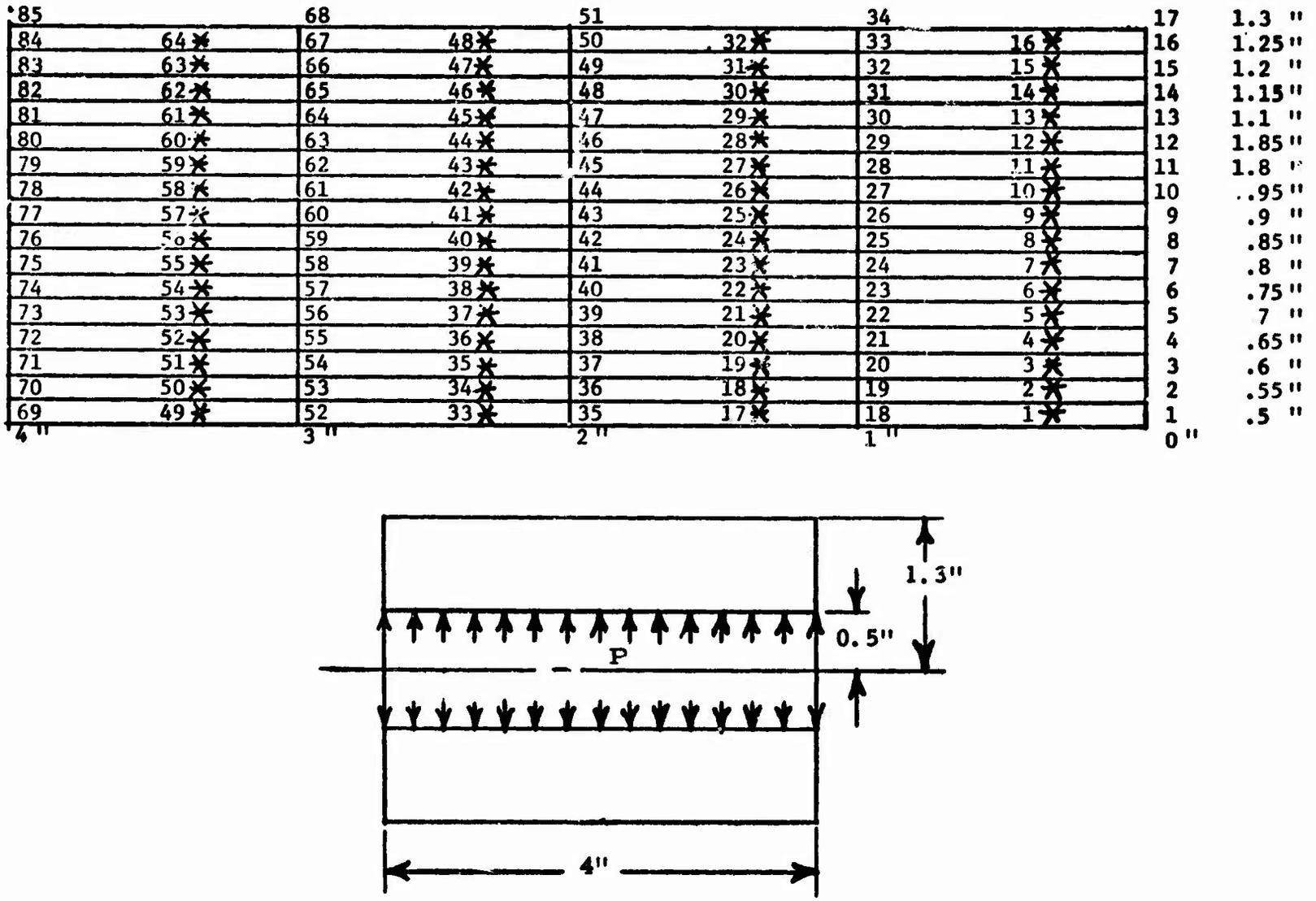


FIGURE 3 NASTRAN MODEL, THICK WALL CYLINDER, PIECE-WISE LINEAR ANALYSIS

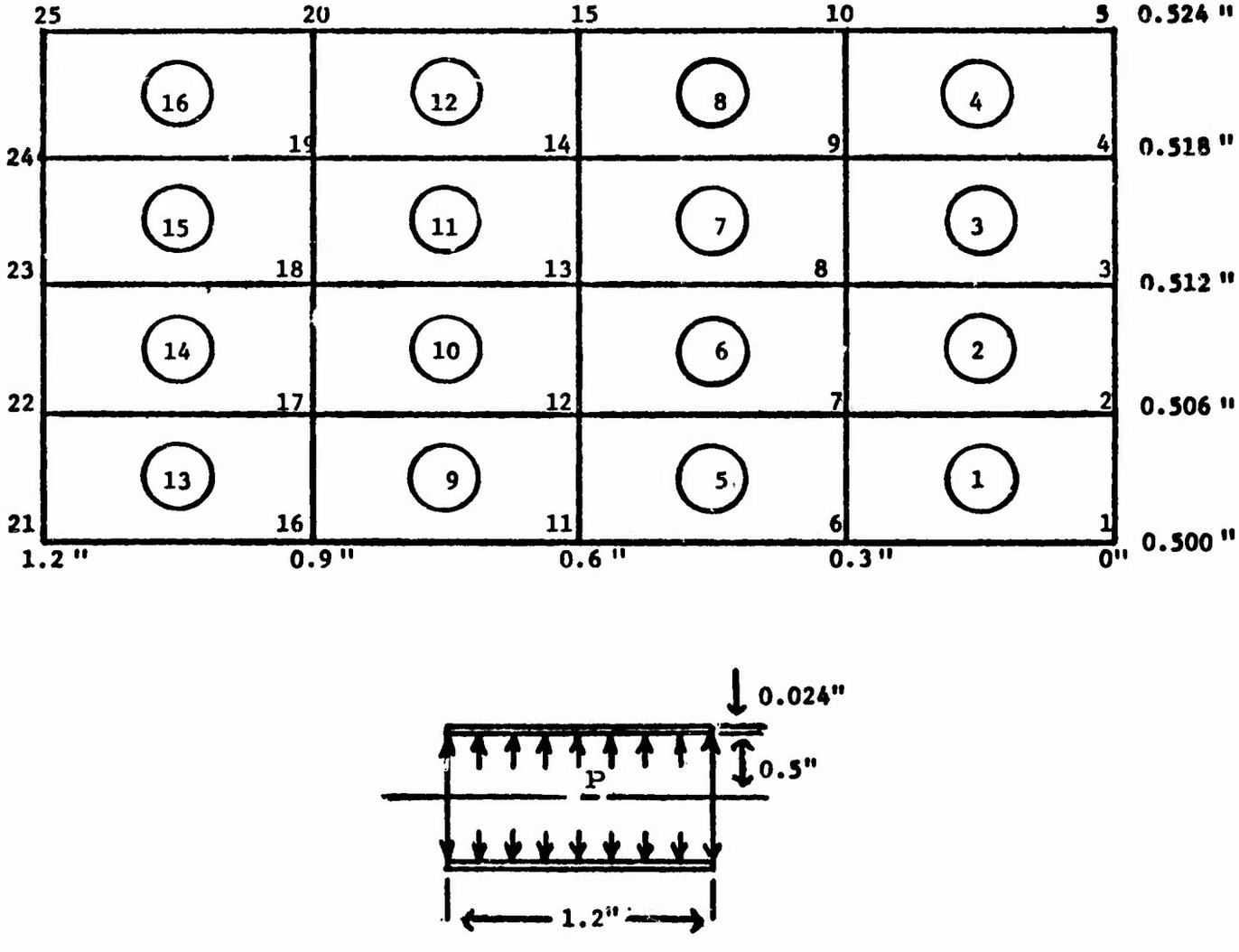


FIGURE 4 NASTRAN MODEL, THIN WALL CYLINDER, PIECE WISE LINEAR ANALYSIS

MATERIAL	RATIO	ELEMENTS
STEEL	2:5	32
	1:5	64
	1:10	128
	1:20	64
	1:10	32
	1:5	16
	1:40	128
	1:80	256
	1:160	128
BRASS	1:1	64
	1:50	32
	1:100	32

FIGURE III
ASPECT RATIO STUDY

FIGURE 5

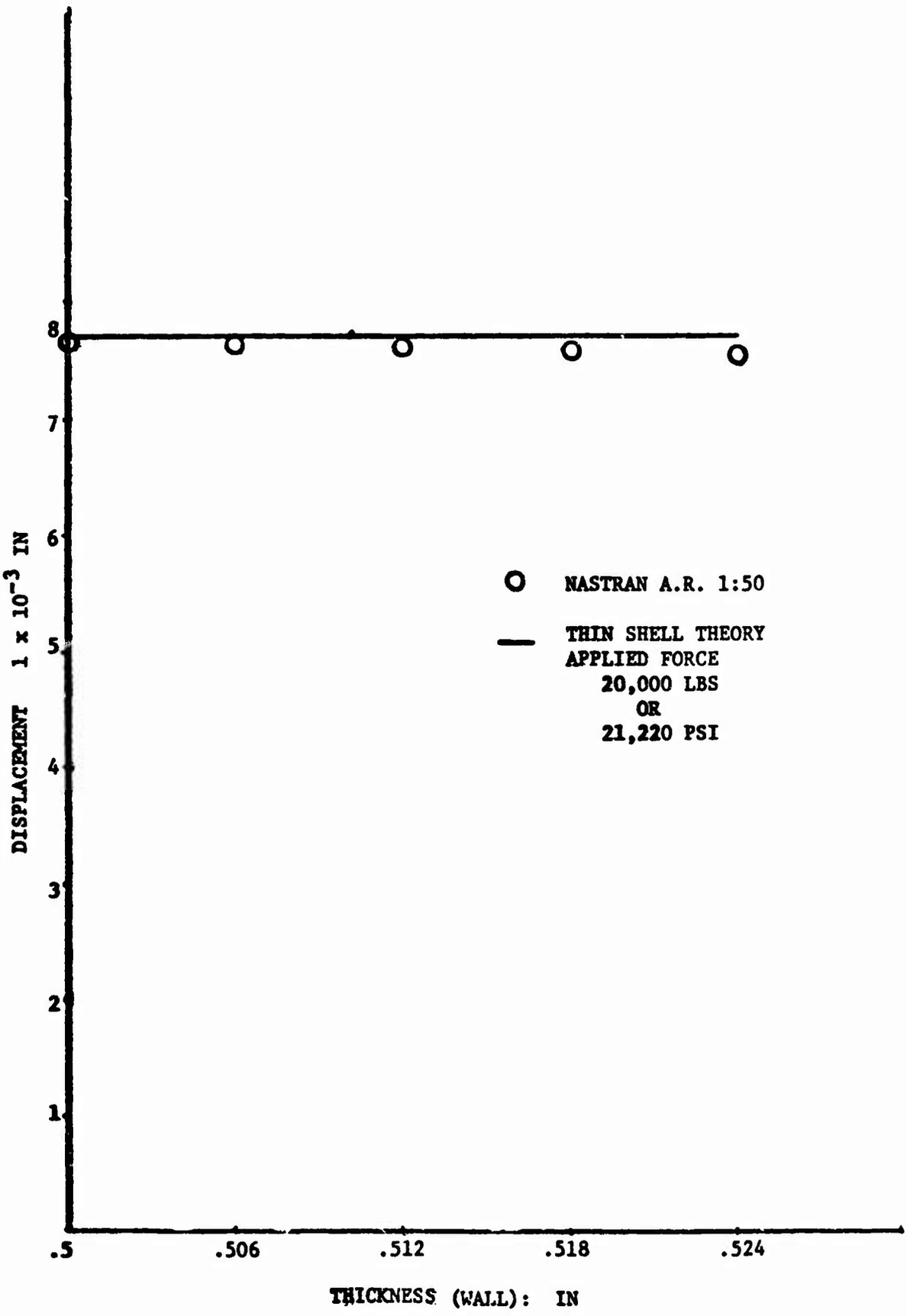

8
7
6
5
4
3
2
1
DISPLACEMENT 1 x 10^{-3} IN
NASTRAN A.R. 1:50
THIN SHELL THEORY
APPLIED FORCE
20,000 LBS
OR
21,220 PSI
.5
.506
.512
.518
.524
THICKNESS (WALL): IN

FIGURE 6 394

ASPECT RATIO	ELEMENTS
1:25	32
1:50	16
1:100	32

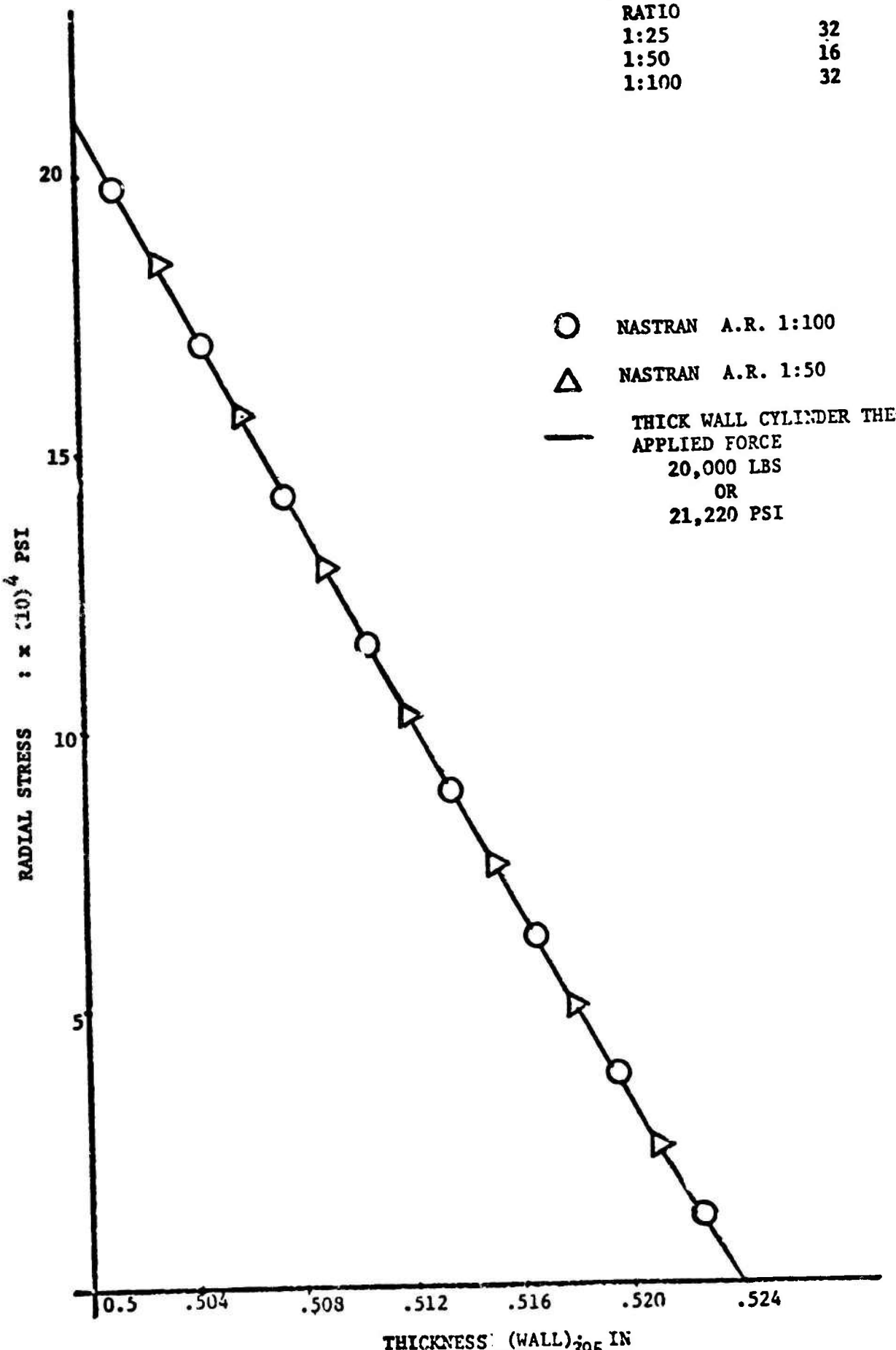


FIGURE 7

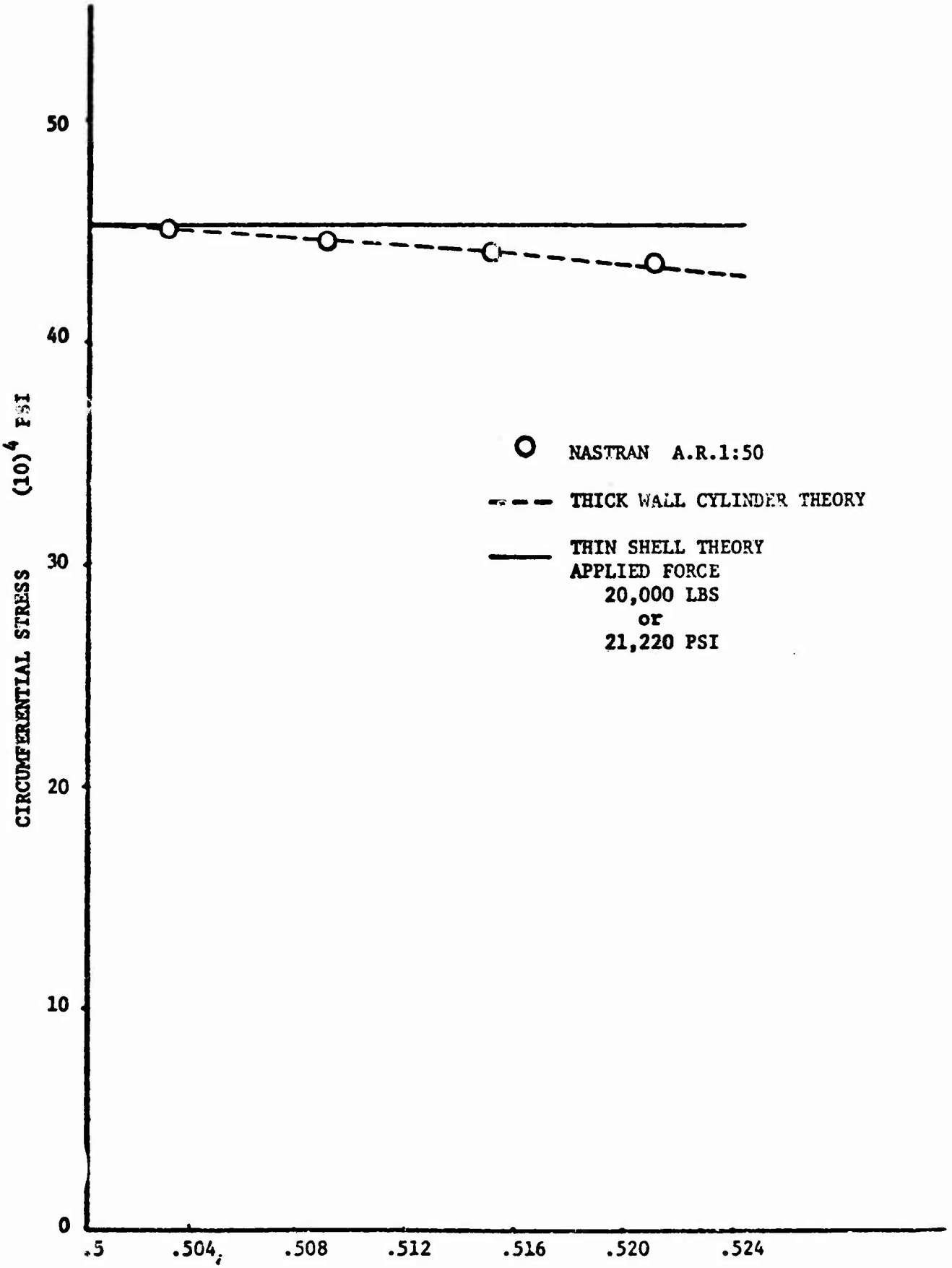
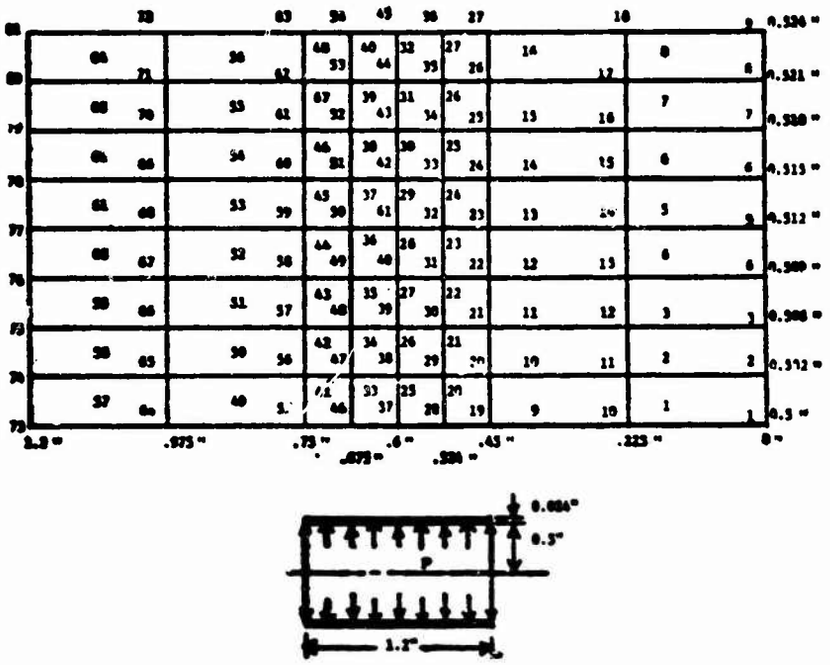

50
40
30
20
10
0
CIRCUMFERENTIAL STRESS (10)^4 PSI
NASTRAN A.R.1:50
THICK WALL CYLINDER THEORY
THIN SHELL THEORY
APPLIED FORCE
20,000 LBS
or
21,220 PSI
.5
.504
.508
.512
.516
.520
.524
THICKNESS (WALL): IN

FIGURE 8



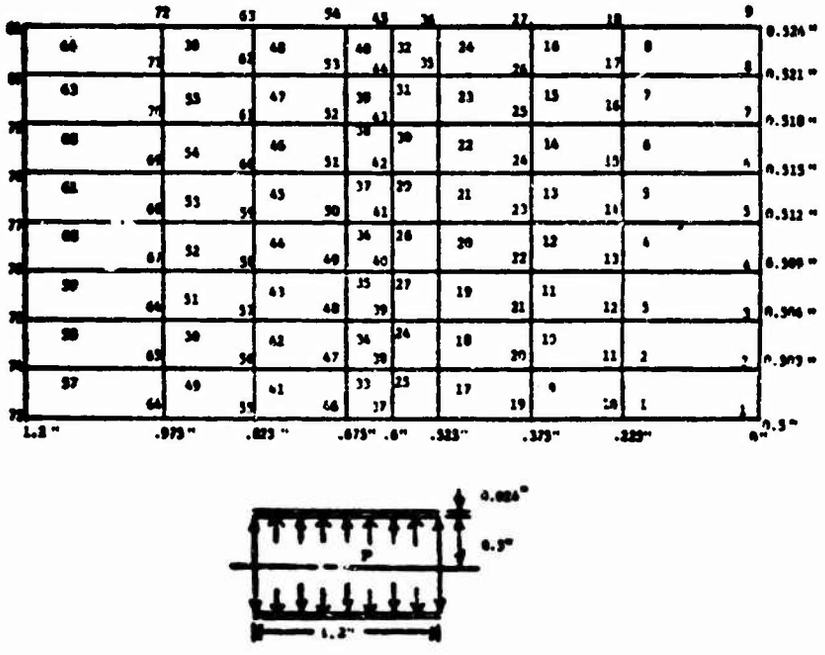


FIGURE 9 NASTRAN MODELS, THIN WALL CYLINDER, ELEMENT SIZE STUDY

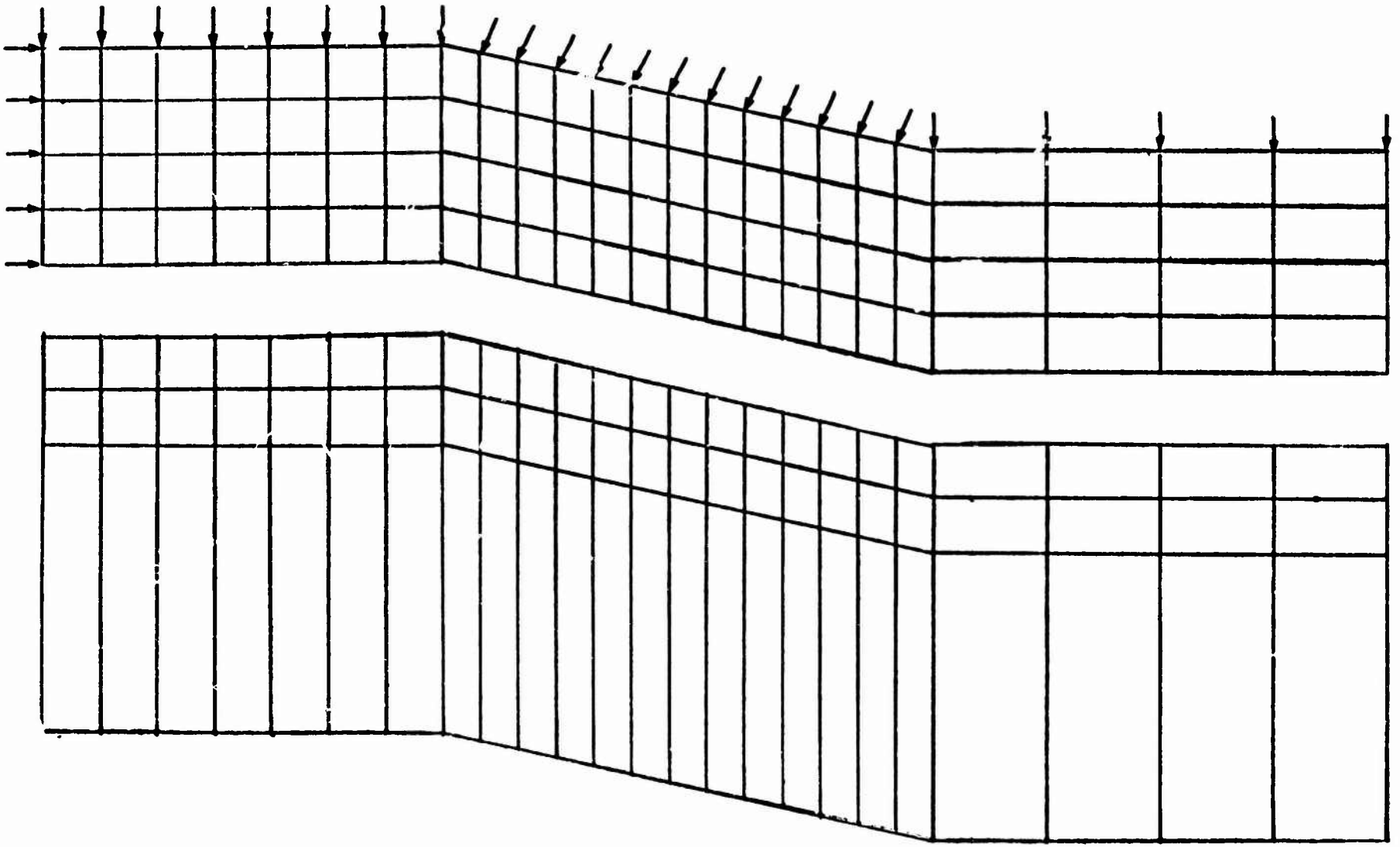


FIGURE 10 CASE CHAMBER INTERFACE MODEL

Cartridge Case Interface Problem

Flow Chart

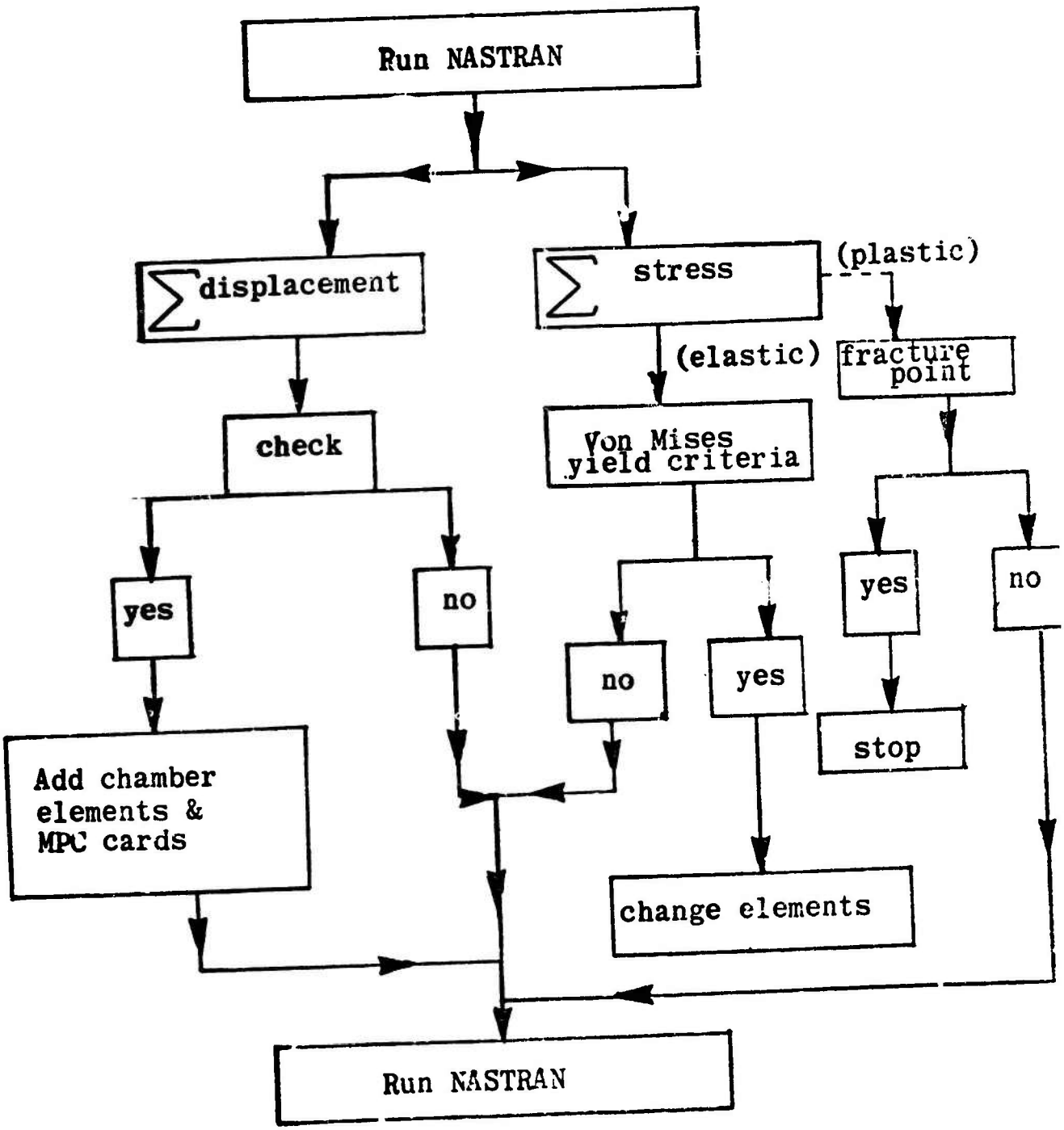


FIGURE 11

A COMPUTERIZED ALGORITHM FOR CALCULATING THE DYNAMIC RESPONSE OF CONTINUA

Paul F. Gordon
Materials Engineering Division
Pitman-Dunn Laboratory
U.S. Army, Frankford Arsenal
Philadelphia, Pa 19137

ABSTRACT. The capability to calculate the dynamic response of continua to highly transient loading environments is an important need in the development and characterization of materials for Army use. In order to meet this need a computer program, HEMP, developed under AEC auspices, has been made operational at Frankford Arsenal. It is the purpose of this paper to present the sequential HEMP-type algorithm which numerically solves the nonlinear partial differential equations relevant to the dynamic response of materials of interest to the Army. The algorithm is a first order accurate finite difference scheme in two space variables and time. The spacial gridwork is fixed in a Lagrangian framework and the time step is determined by explicit satisfaction of a generalized stability criterion for descritized hyperbolic systems.

This algorithm was shown to represent optimal trade-off between: the increased accuracy available from other schemes, generality of material models and simplicity of coding necessary to model real multi-material systems. This result was determined by a comparison between the modular computation scheme and experience with the numerical method of characteristics and other higher order differencing schemes. Modification of the algorithm to include static equilibrium problems is presented.

1. INTRODUCTION. The objectives or topics to be covered in this presentation are shown below (Slide 1)

OBJECTIVES

- HEMP'S EXPLICIT INTEGRATION SCHEME
- APPLICATION TO PROJECTILE IMPACT PROBLEMS
- QUASI-STATIC EQUILIBRIUM PROBLEMS
- NONLINEAR CONSTITUTIVE BEHAVIOR

A brief description of the HEMP[1] code is shown below (Slide 2)

HEMP IS:

- A CODE TO IMPLEMENT FINITE DIFFERENCE APPROXIMATIONS TO THE CONSERVATION RULES (MASS, MOMENTUM, ENERGY) FOR A CONTINUUM IN MOTION
- USED TO ESTIMATE THE STATES OF STRESS, STRAIN, DISPLACEMENT AND VELOCITY IN A BODY (SOLID, GAS, LIQUID) WHOSE MOTION IS INHERENTLY TOW-DIMENSIONAL AND INERTIA DEPENDENT

HEMP HAS:

- A LAGRANGIAN GRIDWORK (FOLLOWS DEFORMING MATERIAL)
- A DIFFERENCE SCHEME WHICH IS EXPLICIT AND USES FICTIVE VISCOSITY TO DAMPEN SHOCK FORMATION
- A MULTI-MATERIAL, MULTI-STATE CAPABILITY
- SPECIAL ROUTINES FOR EXPLOSIVE DETONATION BURNING
- BEEN USED AT FRANKFORD ARSENAL FOR PROJECTILE IMPACT SIMULATION

Because of the versatility of the integration scheme and coding, the HEMP code has a large range of potential application to Frankford Arsenal's missions. A few of these are (Slide 3)

POTENTIAL APPLICATIONS

I. <u>DYNAMIC SIMULATION</u>

- BURSTING/FRAGMENTATION OF SHELL
- CONTAINED EXPLOSIONS (NUCLEAR/NON-NUCLEAR)
- CHARACTERIZING EXPLOSIVE/BURNING CAPABILITY OF PROPELLANT CHARGES IN CARTRIDGE CASES
- SIMULATING TESTING TECHNIQUES FOR METALS AT HIGH STRAIN RATES
- METAL FORMING OPERATIONS: EXPLOSIVE FORGING, WELDING, EXTRUDING, TOOL CUTTING, CHIP FORMATION
- SHAPED-CHARGE FORMATION

II. DYNAMIC STRESS ANALYSIS

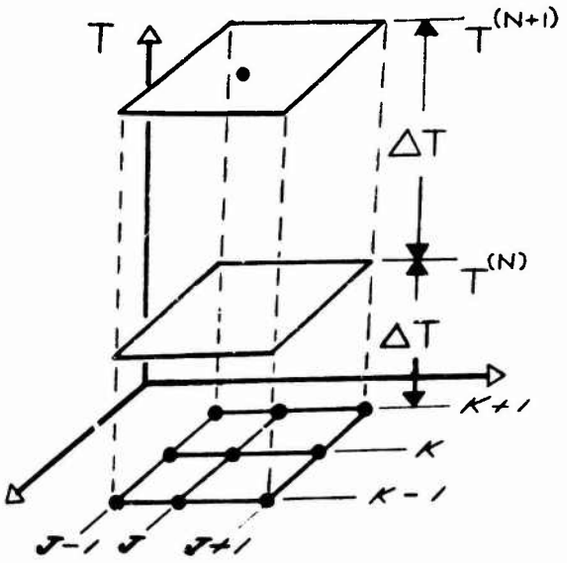
. PROJECTILE/TARGET IMPACT AND PENETRATION

. TRANSIENT LOADING OF STRUCTURES

. ENERGY DEPOSITION ON MATERIALS: LASER, X-RAY

. FUSED PROJECTILES

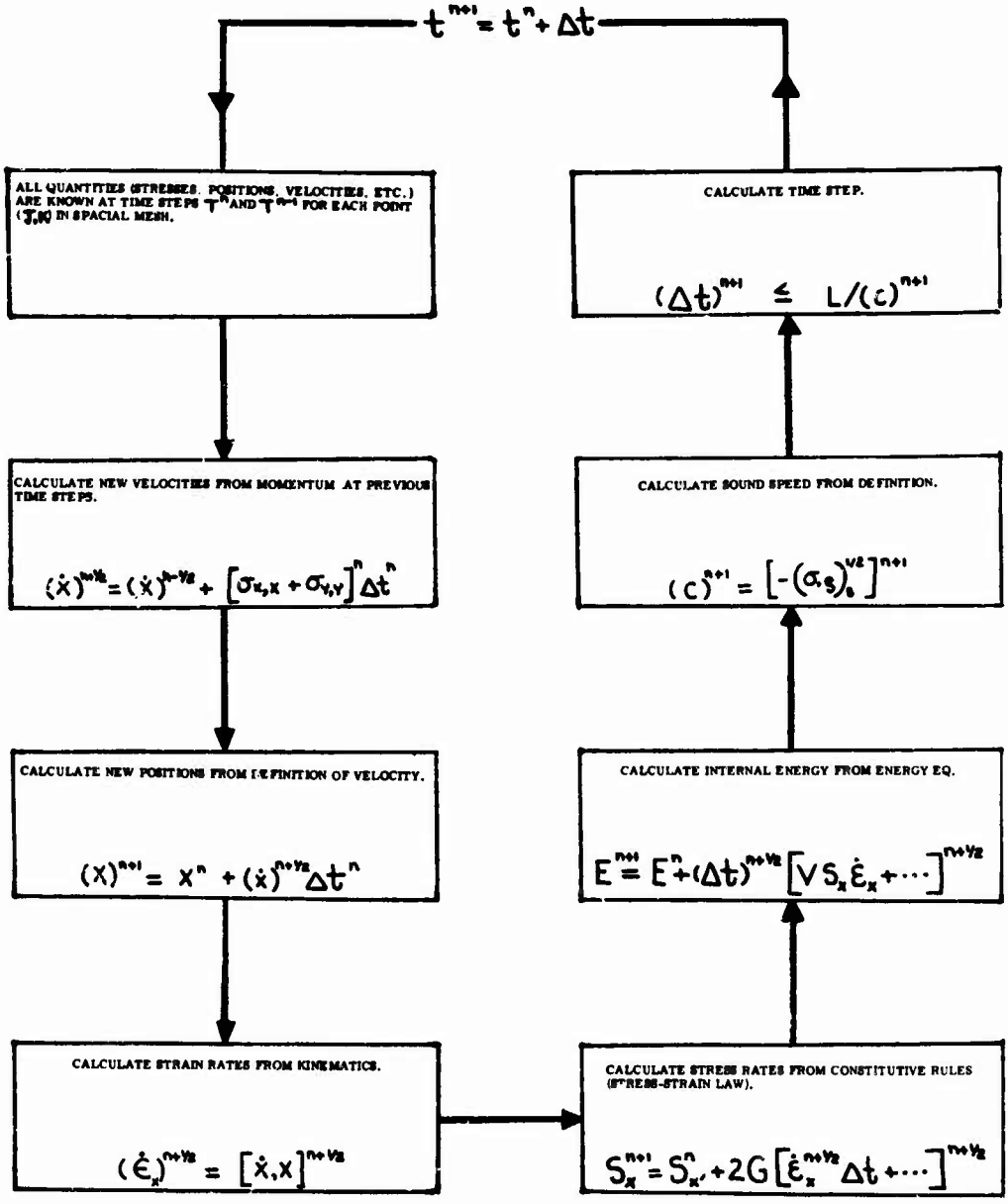
2. INTEGRATION SCHEME. A graphical representation showing the temporal portion of the integration scheme is given below (Slide 4)



TIME INTEGRATION SCHEME. ALL QUANTITIES ARE KNOWN AT $T^{(N-1)}, T^{(N)}$. WE WISH TO CALCULATE ALL QUANTITIES AT $T^{(N+1)}$

In this figure, $T^{(N)}$, $T^{(N+1)}$ and ΔT are, respectively, the times at steps N, N + 1, and the time step. The coordinates J and K are labels used to identify points on a given time plane (e.g., $T^{(N-1)}$ plane). Information has been previously computed and is stored at each of these points on the time planes $T^{(N)}$ and $T^{(N-1)}$. The integration scheme then allows information at $T^{(N+1)}$ to be calculated using the information available from planes $T^{(N)}$ and $T^{(N-1)}$.

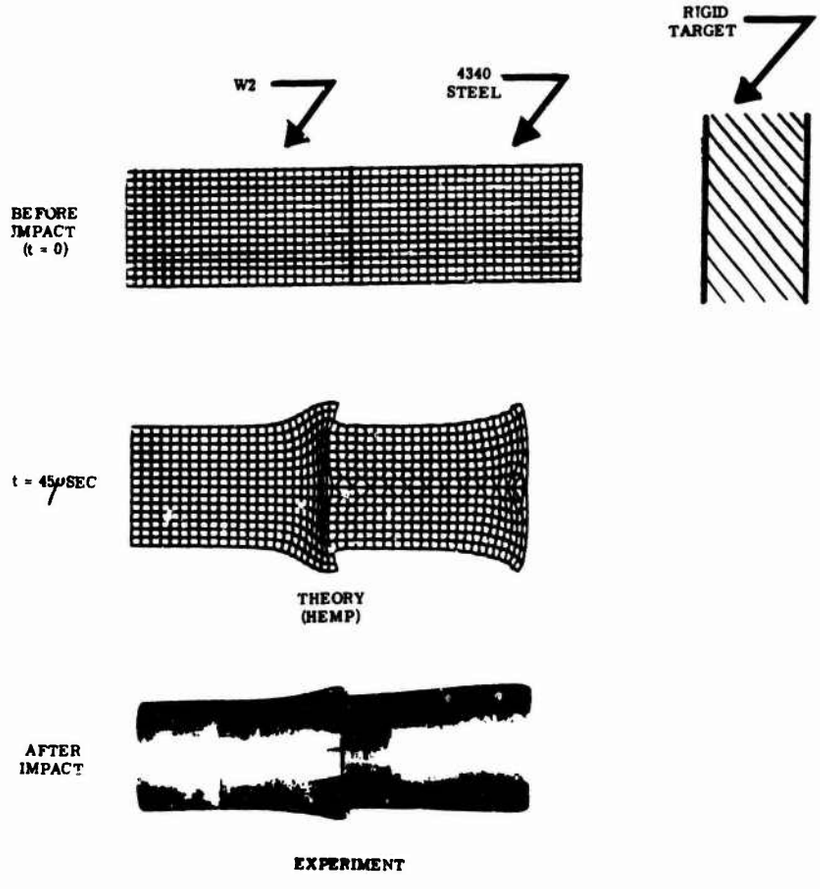
A detailed diagram showing the sequence of calculations is shown below (Slide 5)



CURRENT SEQUENCE OF CALCULATIONS BY TIME STEPS

The most important, and crucial, conclusion to be drawn is that only two of the blocks rely on a specific constitutive description, the others rely on continuum definitions valid for any material. Thus, if the two material blocks (stress rate and energy calculations), which currently are for elastic-plastic, non heat and conducting materials, are replaced by any other valid description, the code is still operational. Some examples are presented in Section 5. Verification of the code for a wide range of problems is presented in Reference (2).

3. APPLICATION TO IMPACT PROBLEMS. One of the applications of HEMP has been to impact problems. Shown below in Slide 6 a post mortem of the normal impact of a composite projectile traveling with a velocity of 731 ft/sec with a thick (rigid) steel target.(3) Shown also is the comparison with a HEMP calculation. The initial gridwork (marked t = 0) is also shown.



COMPARISON OF HEMP PREDICTION VS. EXPERIMENT FOR COMPOSITE PROJECTILE IMPACT ON RIGID TARGET AT 731 FT/SEC.

The comparison illustrates two points: the code performed an excellent simulation of the overall deformation process, and the code is applicable for some types of composite materials.

4. QUASI-STATIC EQUILIBRIUM PROBLEMS. In addition to truly dynamic calculations, HEMP-like codes have been modified to do quasi-static problems as well.[4,5] Shown below (Slide 7) is the stress diffusion or quasi-static concept of Reference (4).

STRESS DIFFUSION CONCEPT

IN DYNAMICS, WE EQUATE THE UNBALANCED FORCE, F_x FROM STRESS EQUILIBRIUM

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_y}{\partial y} = F_x$$

TO THE ACCELERATION (INERTIA) OF THE MEDIUM

$$F_x = \rho \frac{\partial^2 x}{\partial t^2}$$

AT THE END OF EACH TIME STEP. THE RESULTING EQUATIONS ARE PRINCIPALLY HYPERBOLIC. (STRESS WAVES PROPAGATE AND ARE REFLECTED FROM BOUNDARIES).

FOR QUASI-STATICS INTRODUCE A STRESS DIFFUSION TERM

$$F_x = \rho \frac{\partial x}{\partial t^*}$$

IN PLACE OF THE INERTIA TERM.

THE GOVERNING EQUATIONS BECOME PARABOLIC. SUCCESSIVE DISPLACEMENTS TEND TOWARD EQUILIBRIUM.

$$F_x \longrightarrow 0$$

For dynamic problems the unbalanced force on a grid element, Fx, is equated according to Newton's law to acceleration. The result is a hyperbolic system exhibiting wave propagation. If, however, the force is (unrealistically) equated instead to a certain velocity or stress diffusion term, then, as time progresses, Fx tends toward zero. This has the net result of; enforcing static equilibrium, and rendering the equations parabolic. The quantity, τ^*, in Slide 7 is an artificial time variable.

Some quasi-static problems to which the above procedure may be amenable are shown below (Slide 8)

- PROBLEMS IN WHICH INERTIA TERMS, $\rho\ddot{X}$, ARE NEGLIGIBLE

TIME, HOWEVER, IS STILL IMPORTANT FOR PROBLEMS SUCH AS:

- PLASTIC CREEP OF METALS AT ELEVATED TEMPERATURES IN WEAPONS COMPONENTS
- TRANSIENT THERMAL LOADS ON WEAPON STRUCTURES (CARTRIDGE CASES)
- INTERNALLY VISCOUS STRUCTURES SUCH AS PLASTICS OR POLYMERS

5. NONLINEAR CONSTITUTIVE EQUATIONS. Because of the versatility of the HEMP integration scheme (Section 2), constitutive relations other than elasto-plastic can be proposed. Andrews, et. al.[4,5] and Cristescu[6] have presented these in detail. The next two slides (9 and 10) show two examples, in order of increasing complexity, of constitutive equations whose final descritized form are acceptable to HEMP

EXAMPLE "ALMOST EXPLICIT" CONSTITUTIVE EQUATIONS

I. VISCOELASTIC MATERIAL (MAXWELL):PLASTICS, POLYMERS, METALS

$$ds_{ij} = 2G\,d\varepsilon_{ij} - dt\, s_{ij}/\tau$$

τ = RELAXATION TIME

A SECOND ORDER DIFFERENCE ANALOG IS

$$(s_{ij})^{n+1} = (s_{ij})^{n} + 2G\Delta t\,(\dot{\varepsilon}_{ij})^{n+1/2} + -\tfrac{1}{2}\left[(s_{ij})^{n+1} + (s_{ij})^{n}\right]\Delta t/\tau$$

OR

$$(s_{ij})^{n+1} = \left[(s_{ij})^{n} + 2G(\dot{\varepsilon}_{ij})^{n+1/2}\Delta t + -\tfrac{1}{2}(s_{ij})^{n}\Delta t/\tau\right] \Big/ \left[1 + \tfrac{1}{2}\frac{\Delta t}{\tau}\right]$$

WHICH IS EXPLICIT AND LINEAR.

II. VISCOPLASTIC MATERIALS (PERZYNA):METALS

$$ds_{ij} = 2G\,\varepsilon_{ij} + dt\,\gamma\,\Phi(F)\,s_{ij}/\sqrt{I_2^{m}}$$

It can be seen that the Maxwell material leads to a simple explicit formula between the stress at $T(N + 1)$ and the stresses and strain rates at $T(N)$ and $T(N + 1/2)$. (Slide 10)

k, γ = MATERIAL CONSTANTS

$I_s^{(2)}$ = INVARIANT = $S_{ij} S_{ij}$

$F = \sqrt{I_s^{(2)}}/k - 1$

$\Phi(F)$ = MATERIAL FUNCTIONAL)

A SECOND ORDER DIFFERENCE ANALOG IS

$$(S_{ij})^{n+1} = (S_{ij})^n + 2G(\dot{\varepsilon}_{ij})^{n+1/2}\Delta t + - \Delta t \gamma G\left[(\dot{\varepsilon}_{ij})^{n+1} + (S_{ij})^n\right]\Phi/\sqrt{I_s^{(2)}}$$

OR

$$(S_{ij})^{n+1} = (S_{ij})^n\left[1 - \frac{\Delta t \gamma \Phi G}{\sqrt{I_s^{(2)}}}\right] / \left[1 + \frac{\Delta t \gamma \Phi G}{\sqrt{I_s^{(2)}}}\right] + + 2G(\dot{\varepsilon}_{ij})^{n+1/2}\Delta t$$

WHICH IS NONLINEAR BUT "ALMOST EXPLICIT" IF WE ITERATE FOR $I_s^{(2)}$ USING THE AVERAGE

$$\frac{1}{2}\left[S_{ij}^{(n+1)} + S_{ij}^{(n)}\right]$$

In the next example, a viscoplastic solid, the stress at $T(N + 1)$ requires a similar knowledge of two previous time plane calculations. However, because of the non-linearity in $I_s^{(2)}$ and Φ, an iteration is required. In both examples above, the equations must be supplemented by elastic laws, yield criteria, etc.

6. CONCLUSIONS. Based on the above study it can be concluded that:

a. HEMP is adequate for some impact problems involving composite penetrators.

b. The HEMP-type integration scheme is flexible enough to allow viscoelastic and nonlinear viscoplastic material modeling.

c. It is possible with Lagrangian codes to solve some types of quasi-static problems with time dependent boundary conditions.

REFERENCES

1. Wilkins, M. L., "Calculation of Elastic-Plastic Flow", in Methods of Computational Physics, Vol 3, ed. by Alder, et. al., Academic Press, 1964.

2. Karpp, R. R., "Accuracy of HEMP Code Solutions", U.S. Army Ballistic Research Lab Report, M.R. No. 2268, Jan 1973.

3. Schwartz, M. and Sanday, S. C., "Composite Material Projectile Design", Frankford Arsenal Report (in preparation).

4. Andrews, D. J. and Hancock, S. L., "A Relaxation Method for Solving Nonlinear Stress Equilibrium Problems", Journal of Comp. Physics, Vol 12, pp 202-209, 1973.

5. Andrews, D. J., "A Numerical Method for Creep Deformation of Solids", Journal of Comp. Physics, Vol 12, pp 275-279, 1973.

6. Cristescu, N., Dynamic Plasticity, North-Holland Publ. Co., Amsterdam, pp 559-579, 1967.

COMPUTER MODELING IN DETERMINING STABILITY OF A MORTAR REPOSITIONING NONLINEAR CONTROL SYSTEM

C. N. Shen and G. W. Woods
Benet Weapons Laboratory
U. S. Army Armament Command
Watervliet Arsenal
Watervliet, New York

ABSTRACT. The mortar repositioning device is a nonlinear on-off control system with hysteresis, deadzone and dry friction. Stability of the system depends on the existence of a limit cycle with sustained oscillation. This paper gives the piecewise-linear analytic solutions of a mathematical model for the mortar repositioning system with arbitrary starting conditions. The limit cycle sustained oscillation is determined by using a digital computer for the various non-dimensional physical parameters. From the computer results, one can calculate the amount of necessary deadzone in compensating the effects of hysteresis and dry friction to avoid sustained oscillations. The optimal design of deadzone in stabilizing this nonlinear control system is also discussed.

1. INTRODUCTION. A piece-wise linear system behaves like a nonlinear system if the magnitude of the input is a function of the state, such as an on-off system with hysteresis, deadzone and dry friction. The hysteresis is a multiple value function depending on the error signal magnitude and its directions. The dry friction has a constant magnitude and a polarity against the direction of motion. Since all these inputs are state variable dependent it is reasonable to determine the inputs of these piece-wise linear systems by locating them in a phase plane. The phase plane has the error signal as its abscissa and the error rate as its ordinate. One may start a linear computation and end at the boundary of one region in the phase plane. Then the input of the next region is determined before the computation carries on in the second region. In this paper the analytic linear solutions are given with different inputs for various regions. The computer selects the correct inputs in these regions and carries on the computation from region to region, with particular attention to the boundaries values of these regions.

An unstable linear system is unbounded with increase of time while an asymptotic stable linear system approaches a constant. However, the piece-wise linear system, similar to a nonlinear system, may exhibit a sustained oscillation with a limit cycle, which does not exist in a linear system. This limit cycle is not desirable for many engineering applications. To eliminate this limit cycle one can introduce a nonlinear device such as deadzone, to compensate the effect due to hysteresis. The amount of deadzone must be as small as possible because it affects the

the resolution of the output. This paper synthesizes the minimum amount of deadzone required to compensate the effect due to hysteresis and dry friction for the system stability, by means of the results from a digital computer.

2. PIECE-WISE LINEAR ANALYSIS. The nonlinear elements, such as hysteresis and deadzone for a fluidic mortar repositioning control system are given in Appendix A. The mortar dynamics and dry friction, together with the control by pneumatic piston and cylinder are shown in Appendix B.

2.1 System Dynamics. Equations (B-1) through (B-26) give the physical engineering system which leads to the following condensed mathematic form in equations (1) to (9)

$$\frac{d^2x}{d\sigma^2} + \frac{dx}{d\sigma} + \frac{q}{KM} = 0 \tag{1}$$

or

$$x'' + x' + \frac{q}{KM} = 0 \tag{2}$$

where

$$x' = \frac{dx}{d\sigma} \tag{3}$$

with the initial conditions x_0 and x_0. The values of q/KM in equation (1) and its regions in the phase plane are given as follows:

Region number	q/KM	Region of x in phase plane	Region of x' in phase plane	Time Non-dimensional	Equation Number
(1)	$\frac{q_1}{KM} = 1 + \frac{F}{KM}$	$\frac{\Delta+h}{2P} < x$	$0 < x'$	$0 \le \sigma \le \sigma_1$	(4)
(2)	$\frac{q_2}{KM} = 1 - \frac{F}{KM}$	$\frac{\Delta-h}{2P} < x$	$x' < 0$	$\sigma_1 \le \sigma \le \sigma_a$	(5)
(3)	$\frac{q_3}{KM} = 0 - \frac{F}{KM}$	$\frac{-\Delta-h}{2P} < x < \frac{\Delta-h}{2P}$	$x' < 0$	$\sigma_a \le \sigma \le \sigma_b$	(6)
(4)	$\frac{q_4}{KM} = -1 - \frac{F}{KM}$	$x < \frac{-\Delta-h}{2P}$	$x' < 0$	-	(7)
(5)	$\frac{q_5}{KM} = -1 + \frac{F}{KM}$	$x < \frac{-\Delta+h}{2P}$	$0 < x'$	-	(8)

Region number	q/KM	Region of x in phase plane	Region of x' in phase plane	Time Non-dimensional	Equation Number
(6)	$\frac{q_6}{KM} = 0 + \frac{F}{KM}$	$\frac{-\Delta+h}{2P} < x < \frac{\Delta+h}{2P}$	$0 < x'$	-	(9)

The above relationship is shown by the phase plane in Figure 1. Note that equation (1) is non-dimensionalized and the values of q/KM become either 1, 0, or -1 if there is no dry friction (F = 0) in equations (4) - (9).

2.2 The Analytic Solution for Trajectories in Phase-Plane. For any starting conditions x_o and x_o' in region (1) when $0 \leq \sigma \leq \sigma_1$ the analytic solution for equation (1) is

$$x' = x_o' e^{-\sigma} + \frac{q_1}{KM} (e^{-\sigma} - 1) \tag{10}$$

$$x = x_o - \frac{q_1 \sigma}{KM} + (x_o' + \frac{q_1}{KM}) (1 - e^{-\sigma}) \tag{11}$$

When $\sigma = \sigma_1$ the region ends at

$$x' = x_1' = 0 \tag{12}$$

Thus

$$0 = x_o' e^{-\sigma_1} + \frac{q_1}{KM} (e^{-\sigma_1} - 1) \tag{13}$$

from which one obtains

$$\sigma_1 = \ell n[(x_o' + \frac{q_1}{KM}) / (\frac{q_1}{KM})] \tag{14}$$

From equation (11) we have

$$x_1 = x_o - \frac{(q_1)}{KM} \sigma_1 + x_o' \tag{15}$$

where q_1/KM is given in equation (4).

The end conditions (12) and (15) for region (1) become the initial conditions of equation (1) in region (2) when $\sigma_1 \leq \sigma \leq \sigma_a$. Thus

$$x' = x_1' e^{-(\sigma-\sigma_1)} + \frac{q_2}{KM} [e^{-(\sigma-\sigma_1)} - 1] \tag{16}$$

$$x = x_1 - \frac{q_2}{KM}(\sigma - \sigma_1) + (x_1' + \frac{q_2}{KM})[1 - e^{-(\sigma-\sigma_1)}] \quad (17)$$

When $\sigma = \sigma_a$ the region ends at

$$x_a = (\Delta - h)/(2P) \quad (18)$$

Let the initial condition of x at region (1) be

$$x_o = (\Delta + h)/(2P) \quad (19)$$

By using equations (17) - (19) we can determine $\sigma_a - \sigma_1$ by

$$-\frac{h}{P} = (x_1 - x_o) - \frac{q_2}{KM}(\sigma_a - \sigma_1) + (\frac{q_2}{KM})[1 - e^{-(\sigma_a - \sigma_1)}] \quad (20)$$

where $x_1 - x_o$ is given in equation (15)

From equations (12) and (16) the final condition of x_a' becomes

$$x_a' = \frac{q_2}{KM}[e^{-(\sigma_a - \sigma_1)} - 1] \quad (21)$$

where q_2/KM is given in equation (5).

The end conditions (18) and (21) for region (2) become the initial conditions of equation (1) in region (3) when $\sigma_a \leq \sigma \leq \sigma_b$. Thus

$$x' = x_a' e^{-(\sigma-\sigma_a)} + \frac{q_3}{KM}[e^{-(\sigma-\sigma_a)} - 1] \quad (22)$$

$$x = x_a - \frac{q_3}{KM}(\sigma - \sigma_a) + (x_a' + \frac{q_3}{KM})[1 - e^{-(\sigma-\sigma_a)}] \quad (23)$$

When $\sigma = \sigma_b$ the region ends at

$$x_b = (-\Delta - h)/(2P) \quad (24)$$

From equations (18), (23) and (24) we can determine $\sigma_b - \sigma_a$ by

$$-\frac{\Delta}{P} = -\frac{q_3}{KM}(\sigma_b - \sigma_a) + (x_a' + \frac{q_3}{KM})[1 - e^{-(\sigma_b - \sigma_a)}] \quad (25)$$

after which the final condition of x_b' becomes

$$x_b' = x_a' e^{-(\sigma_b - \sigma_a)} + \frac{q_3}{KM}[e^{-(\sigma_b - \sigma_a)} - 1] \quad (26)$$

where q_3/KM is given in equation (6).

Equations (24) and (26) are the final conditions of region (3).

Since the system is symmetric about the origin in Figure 1, the analytic solution for regions (4), (5) and (6) is similar to the regions (1), (2) and (3). One can derive these equations without difficulty.

From these non-dimensional equations, a similarity among equations (10), (16), and (22) and among (11), (17), and (23) exists. Equations (10), (16), and (22) can be written in the general form

$$x' = x'_A \exp(\sigma_a - \sigma) + Q[\exp(\sigma_a - \sigma) - 1] \qquad (27)$$

and equations (11), (17), and (23) in the general form

$$x = x_A - Q(\sigma - \sigma_a) + (x'_A + Q)[1 - \exp(\sigma_a - \sigma)] \qquad (28)$$

where x'_A are initial starting velocities x'_0, x'_1, and x'_a for the regions 1, 2, and 3 respectively, x_A are initial starting positions x_0, x_1, and x_a for the regions 1, 2, and 3 respectively, and Q are q_1/KM, q_2/KM, and q_3/KM for the regions 1, 2, and 3 respectively.

3. COMPUTER PROGRAMMING FOR THE PIECE-WISE LINEAR ANALYTICAL SOLUTIONS

3.1 Generalizing the Non-dimensional Equations for use in the Computer.

Equations (27) and (28) are transformed into the computer software language of Fortran IV as seen below in equations (29) and (30).

```
Y(I) = YA*Z + Q*(Z-1).                          (29)

X(I) = XA - Q*(S-SA) + (YA+Q)*(1-Z)             (30)
```

The variables as set up in the computer equations (29) and (30) relate to the non-dimensional equations as follows:

1) The time variable σ becomes S, and σ_1, σ_a, etc. become SA where SA takes on new values for each new region of the phase plane.
2) The quantity Z = exp (SA-S), where SA = 0 in region 1.
3) The quantity XA is x_0, x_1 or x_a in regions 1, 2, or 3 respectively, where x_0 is the initial starting condition in the X plane for region 1. The quantities x_1, x_a, etc. are the last values of X(I) in the previous regions which are used as initial conditions in the following regions.
4) The quantity YA is x'_0, x'_1 or x'_a, in regions 1, 2 or 3, respectively, where x'_0 is the initial starting condition in the Y plane for region 1. The quantity x'_1, x'_a, etc. are the last

values of Y(I) in the previous regions which are used as initial conditions in the following regions.

5) The quantity Q is q_1/KM, q_2/KM or q_3/KM. The solutions to the equations (29) and (30) are piece-wise linear because Q takes on new values for each of the six regions of the phase plane as shown in equations (4) - (9). These changing values of Q are due to the driving force and the frictional force changing throughout the system. Because of these changes of Q, the differential equations are only linear in each region. At the boundaries of each region in the phase plane the trajectories are continuous in their positions and velocities.

3.2 Critical Points of the Phase Plane Diagram. In the phase plane of figure 1, the critical points at the boundary of two neighboring regions are as follows:

Boundary Region	Critical Point at	
6 and 1	$x_0 = (\Delta+h)/2P$,	$x_0' =$ arbitrary positive
1 and 2	$x_1 =$ arbitrary	$x_1' = 0$
2 and 3	$x_a = (\Delta-h)/2P$	$x_a' =$ arbitrary negative
3 and 4	$x_b = (-\Delta-h)/2P$	$x_b' =$ arbitrary negative
4 and 5	$x_2 =$ arbitrary	$x_2' = 0$
5 and 6	$x_3 = (-\Delta+h)/2P$	$x_3' =$ arbitrary positive

The parameters $\Delta/2P$ and $h/2P$ are known quantities from design and testing of the system and will be discussed further in Chapters 4 and 5.

3.3 The Programming Scheme. The function of this computer program is to generate a phase plane plot when the various parameters $\Delta/2P$, $h/2P$, Q for each region, and the initial conditions, x_0, x_0' are given. The program starts at time (σ or S) = 0 and solves equations (29) and (30) for small, increasing steps of time. The computation of equations (29) and (30) halts when the value of X(I) or Y(I) is within a tolerance of the correct critical boundary point for the region. The program then resumes computing values of X(I) and Y(I) but at a time interval 1/10 of the original time interval until X(I) or Y(I) is within 1/10 of the original tolerance of the critical boundary point. These last calculated values X(I), Y(I), and S are now the new values for XA, YA, and SA respectively. Solving for X(I) and Y(I) at the original time step resumes with the new values in XA, YA, SA and Q for the new region. If the system is unstable, sustained oscillations will result in the phase plane plot.

The program continually checks the value of Y(I) in regions 3 and 6 to see if Y(I) approaches zero from a negative value in region 3, or approaches zero from a positive value in region 6. If Y(I) approaches zero within a given tolerance in either region 3 or 6 before X(I) reaches a critical boundary point, the system is stable. At this point the

program halts further computation and plots the stable phase plane diagram. The computer program and flow chart are given in Appendix C.

4. THE EXISTENCE OF A LIMIT CYCLE FROM COMPUTER SOLUTIONS.

4.1 Definition. The term limit cycle in nonlinear systems refers to limiting values of closed curve trajectories in the phase space corresponding to periodic motion. Limit cycles may be stable or unstable. In figure (2a) all close trajectories approach the limit cycle as time approaches infinity. In figure (2b) all the close trajectories move away from the limit cycle as time goes on, making for an unstable limit cycle.

4.2 The Role of the Computer. The computer is a valuable tool to the engineer who is trying to establish the existence or non-existence of a limit cycle for a system. In this nonlinear control system, the main objective is to obtain a final settling point within a reasonable period of time. Therefore, it is desirable not to have a limit cycle or sustained oscillations in this system since these correspond to stable periodic motion.

4.3 Description of the Computer Plot. Phase plane plots of the variables X(I) and Y(I) as calculated in equations (29) and (30) are shown in figures (3), (4), and (5) with the values of $h/2P = 0.10$ and $F/KM = 0.50$. A limit cycle or sustained oscillation is shown in figure (3). In figure (3), the boundary regions between regions 2 and 3 and between regions 5 and 6 are on the opposite side of the ordinate as compared in figure (1). The reason for this is, if $\Delta/2P<h/2P$, these boundaries are as shown in figure (1), but if $h/2P<\Delta/2P$, the boundaries are as shown in figure (3). In figure (3), regions 3 and 6 are narrow due to the small value for deadzone, $\Delta/2P = 0.028$.

If $\Delta/2P$ is increased to 0.038 as shown in figure (4), the phase plane changes from sustained oscillations, to a trajectory with a final settling point because Y(I) approaches zero in region 3 before X(I) reaches the boundary point $(-\Delta-h)/2P$. This boundary point value is -0.138 whereas the computer value for X(I) = -0.1368 when Y(I) approaches zero. Comparing the closeness of these two numerical values, indicates that the system is just at the margin of stability.

Increasing $\Delta/2P$ still further to 0.040 brings stability to the system, and is shown by the plot in figure (5). Regions 3 and 6 in figure (5) are wider than these regions in figures (3) or (4). Stability is reached sooner in time in figure (5) than in figure (4) when comparing 10.740 seconds in figure (4) to 5.355 seconds in figure (5). A further increase in deadzone would bring about a settling point sooner in time but the final settling point may fall anywhere within the deadzone depending upon the original arbitrary starting conditions of X0 and Y0.

5. THE NECESSARY DEADZONE IN COMPENSATING THE EFFECT OF HYSTERESIS AND DRY FRICTION.

5.1 The System's Deadzone, Hysteresis and Dry Friction. The computer reads in values for the three parameters deadzone ($\Delta/2P$), hysteresis ($h/2P$), and dry friction (F/KM) along with the initial arbitrary starting point XO and YO in order to compute values for X(I) and Y(I). The term F/KM is the only parameter of the three that is present in equations (29) and (30) and is contained in the Q term. The Q term takes on the value of q_i/KM as seen in equations (4) through (9) where, $i = 1,2,3\cdots6$ and corresponds to each of the six phase plane regions. The deadzone and hysteresis form the boundary point values to which X(I) is compared in four of the six regions as shown in the table in section (3.2).

The hysteresis in the system is the inherent property of certain components and very little can be done to alter it's value. Whereas, the value of the dry friction in the system is more subject to change through design and lubrication.

The deadzone is the main adjustable parameter in the system. If, for a given set of system parameters, it is found that the system is unstable, the deadzone may be increased in an attempt to bring about stability.

5.2 Varying the Parameters. The three parameters have systematically been varied to show the effect each parameter has on the system. The following is a list of the twelve cases shown in the phase plane plots of figures (3) through (14).

TABLE I

Figure Number	h/2P	F/KM	$\Delta/2P$	$(\Delta+h)/(2P)$
3	0.10	0.50	0.028	
4	0.10	0.50	0.038	0.138
5	0.10	0.50	0.048	
6	0.10	0.25	0.075	
7	0.10	0.25	0.087	0.187
8	0.10	0.25	0.098	
9	0.027	0.50	0.010	
10	0.027	0.50	0.016	0.043
11	0.027	0.50	0.025	
12	0.027	0.25	0.028	
13	0.027	0.25	0.038	0.065
14	0.027	0.25	0.048	

In the following discussion of Table I, refer to the table for the numerical values of the parameters discussed in each figure. Figures (3), (4) and (5) have h/2P and F/KM fixed, and vary Δ/2P. These three figures were discussed in detail in section (4.3).

In figures (6), (7) and (8) the hysteresis is maintained at its previous value but the dry friction was reduced and the deadzone varies to yield the results of sustained oscillations for figures (6) and (7) and stability or a final settling point for figure (8). Although figure (7) has a sustained oscillation, it is close to the margin of stability since a small increase in deadzone produced stability.

The hysteresis has been decreased for figures (9) through (14). A sustained oscillation resulted from the values of the parameters in figure (9), while figures (10) and (11) show stability. The stability of figure (10) is again marginal while that of figure (11) is definite.

The smaller values for hysteresis and dry friction are shown in figures (12), (13) and (14). Figure (12) shows a sustained oscillation. Figure (13) is a very good example of marginal stability because: upon close examination a final settling point can be seen. Again, a final settling point is reached in figure (14).

5.3 Determining the Margin of Stability. The margin of stability as discussed in section (4.3) is when X(I) approaches (Δ+h)/2P, or its negative value, at the same time that Y(I) is approaching zero. The computer can print out these values of X(I) as Y(I) approaches zero for comparison.

Figures (4), (7), (10) and (13) were stated as being marginally stable. Table II compares the last and next to last values of X(I) when Y(I) approaches zero, to the calculated value of (Δ+h)/2P or its negative value. The last column of Table II is repeated from the last column of Table I.

TABLE II

Figure Number	Value of X(I) as Y(I) ≈ 0 Next to Last Value	Last Value	(Δ+h)/2P
4	0.1388	-0.1368	0.1380
7	-0.1872	0.1869	0.1870
10	-0.0432	0.0410	0.043
13	-0.0651	0.0637	0.065

From this table it can be seen that the last and next to last values of X(I) are very close to the value of (Δ+h)/2P or its negative. Since the initial starting conditions are arbitrary, a convenient starting point when looking for the margin of stability is at the point Y0 = 0,

XO = (Δ+h)/2P. Within a half cycle, the nearness of X(I) to (-Δ-h)/2P as Y(I) approaches zero can be observed and a judgment made as to whether the system is unstable, marginally stable or stable.

5.4 Necessary Deadzone for Marginal Stability. From these twelve cases and twelve other cases not shown, a table of marginal stability can be constructed for this system. The dry friction values of 0.0, 0.25, 0.50 and 0.75 are read in the columns, while the two values of hysteresis, 0.10 and 0.027, are read across the rows. The deadzone value for marginal stability is read at the intersection of the rows and columns. For example: if the system has a dry friction value of 0.25 and a hysteresis value of 0.10, the deadzone value at the point of marginal stability is 0.087. If a final settling point is desired, the deadzone value has to be greater than 0.087. If a value less than 0.087 is chosen, the system will have sustained oscillations.

TABLE III

F/KM	0.00	0.25	0.50	0.75
h/2P				
0.100	0.250	0.087	0.038	0.0105
0.027	0.150	0.038	0.016	0.0055

If from a set of system parameters, the hysteresis or dry friction is changed and the system becomes unstable, the chart above can give the engineer an idea of how much deadzone is needed to compensate for the effect of the changing parameter.

6. CONCLUSION. This report shows how equations of motion for non-dimensional, piece-wise linear equations behave like a nonlinear system. The computer calculates the solutions for these equations and plots the phase plane trajectory for each set of system parameters chosen.

In some cases when writing the equations to plot the phase plane trajectory, time can be eliminated as the independent variable. However, time has remained as the independent variable in this system while calculating the variables X(I) and Y(I). The reason for this is that the response time of the system is important and can be easily recorded.

The computer phase plane plots are valuable in locating the margin of stability. If the initial conditions are chosen at XO = + (Δ+h)/2P) and YO = 0.0, within half a cycle it can be determined if the trajectory is near the margin of stability. If X(I) approaches ± (Δ+h)/(2P) before Y(I) approaches zero, the system will have a sustained oscillation, but if Y(I) approaches zero first, the system is stable upon reaching a final settling point. The margin of stability can be established when X(I) approaches ±(Δ+h)/(2P) and Y(I) approaches zero simultaneously. A table for marginal

stability was constructed from the computer output, and from Table I, the effect of changing the deadzone to stabilize the system can be determined.

The resolution of the system must be kept in mind when varying the parameters. Increased deadzone may give a stable system, but too much deadzone will decrease the resolution of the system.

In this paper it is seen that the computer plays a large role in determining how the parameters of a system can be varied to establish a stable system especially at the margin of stability.

APPENDIX A. The Nonlinear Elements. A sensor is used to measure the relative angular position ε of the mortar barrel with reference to the axis of a cylinder which is mounted on the tripod of the mortar. In the sensor there are two orifices through which the air may pass to the input of the back pressure switch. A deadzone can be created by adjusting the angular position of these orifices in the cam. Thus the two orifices can be arranged such that they are either normally blocked or normally open.

The back pressure switch exhibits the characteristic of a hysteresis. The output of the switch turns on at a high threshold of the input while it turns off at a low threshold.

The output of the back pressure switch is connected to an on-off fluidic amplifier which has practically no hysteresis. The compressed air from the power amplifier pushes the piston of the cylinder for controlling the angular position of the mortar.

For each input angle of rotation ε of the mortar barrel two pressure outputs can be obtained, one on each side of the piston inside the cylinder. This is a push-pull arrangement for an on-off fluidic system. The output-input relationship indicates that a combined hysteresis and deadzone exist as shown in figure (A1).

Figure (A1) plots m, the net output force on piston by gas pressure vs. ε, the relative angle of rotation of the input. The total hysteresis is h and the total deadzone is Δ. At point "a" where ε is $\frac{1}{2}(\Delta+h)$ the output jumps from zero to M. As the input ε increases to point "b" the output is kept at the same amplitude M until the input drops to point "c". The input ε at point "c" is $(1/2)(\Delta-h)$ and the output drops to zero again. As the input further decreases to point "d" at the amount $-(1/2)(\Delta+h)$, the output becomes -M. The second half cycle repeats the same scheme. These are shown as points "d,e,f, and a".

In order to facilitate analysis the phase plane drawing of ε and $\dot{\varepsilon}$ is desirable. This can be achieved by transforming figure (A1) into figure (A2).

There are four threshold values of ϵ in figure (A1). These values are $1/2(\Delta+h)$, $1/2(\Delta-h)$, $-1/2(\Delta-h)$, and $-1/2(\Delta+h)$, corresponding to four points in figure (A1) and four vertical lines in figure (A2). The output m = M for points "a,b and c" in figure (A1) is the shaded region which is marked m = M in figure (A2). The output m is zero for points "c and d" and is so indicated in an unshaded region in figure (A2), which gives the output at any region in the phase plane for a nonlinear element with both hysteresis and deadzone.

APPENDIX B. The Dynamics and Dry Friction. The mortar barrel is rotating while the piston is set in motion by the pressure on one or both sides of the piston as shown in figure (B1). Let the equivalent mass of the piston and its connected part be m_e, the coefficient of viscous damping be μ, the displacement of the piston be c, the net pneumatic pressure on the piston be p, the area of the piston be A and the dry friction be G. The equation of motion becomes for net pneumatic pressure p on the left of the piston.

$$m_e \frac{d^2c}{dt^2} = -\mu \frac{dc}{dt} + pA - G \qquad \text{if } \frac{dc}{dt} > 0 \qquad \text{(B1)}$$

$$m_e \frac{d^2c}{dt^2} = -\mu \frac{dc}{dt} + pA + G \qquad \text{if } \frac{dc}{dt} < 0 \qquad \text{(B2)}$$

For zero net pressure on the piston

$$m_e \frac{d^2c}{dt^2} = -\mu \frac{dc}{dt} - G \qquad \text{if } \frac{dc}{dt} > 0 \qquad \text{(B3)}$$

$$m_e \frac{d^2c}{dt^2} = -\mu \frac{dc}{dt} + G \qquad \text{if } \frac{dc}{dt} < 0 \qquad \text{(B4)}$$

For net pneumatic pressure p on the right of the piston

$$m_e \frac{d^2c}{dt^2} = -\mu \frac{dc}{dt} - pA - G \qquad \text{if } \frac{dc}{dt} > 0 \qquad \text{(B5)}$$

$$m_e \frac{d^2c}{dt^2} = -\mu \frac{dc}{dt} - pA + G \qquad \text{if } \frac{dc}{dt} < 0 \qquad \text{(B6)}$$

where the quantities m_e, μ, p, A and G are all positive real parameters. The above equations indicate that the pneumatic force pA depends on which side of the piston is pressurized and that the dry friction is a function of the direction of the velocity of the piston. Let the error rate of the system be

$$\frac{d\varepsilon}{dt} = -\rho \frac{dc}{dt} \tag{B7}$$

substituting equation (B7) into equation (B1) one obtains

$$-\frac{m_e}{\rho}\frac{d^2\varepsilon}{dt^2} = \frac{\mu}{\rho}\frac{d\varepsilon}{dt} + pA - G \tag{B8}$$

which is equivalent to

$$\tau \frac{d^2\varepsilon}{dt^2} + \frac{d\varepsilon}{dt} + q = 0 \tag{B9}$$

where

$$q = Km_s - f \tag{B10}$$

$$K = \rho/\mu, \tag{B11}$$

$$\tau = m_e/\mu \tag{B12}$$

$m_s = M = pA$ if pressure applies on the left (B13)

$m_s = 0$ if net pressure on the piston is zero (B14)

$m_s = -M = -pA$ if pressure applies on the right (B15)

$f = F = \frac{\rho G}{\mu}$ if $\frac{d\varepsilon}{dt} < 0$ (B16)

$f = -F = -\frac{\rho G}{\mu}$ if $\frac{d\varepsilon}{dt} > 0$ (B17)

Equations (B13) to (B17) are derived based on equations (B1) to (B4).

To summarize the above formulations there are six regions in the phase plane plot in figure B2. The force function q in equation (B9) takes the following values:

Value of $q = Km_s - f$	Net Pressure on Piston	Mortar Position	Relative Mortar Velocity	Equation Number
KM + F	on left	$(\Delta+h)/2<\varepsilon$	$\dot{\varepsilon} > 0$	(B-18)
KM - F	on left	$(\Delta-h)/2<\varepsilon$	$\dot{\varepsilon} < 0$	(B-19)
0 - F	none	$(-\Delta-h)/2<\varepsilon<(\Delta-h)/2$	$\dot{\varepsilon} < 0$	(B-20)
- KM - F	on right	$\varepsilon<(-\Delta-h)/2$	$\dot{\varepsilon} < 0$	(B-21)
- KM + F	on right	$\varepsilon<(-\Delta-h)/2$	$\dot{\varepsilon} > 0$	(B-22)
0 + F	none	$(-\Delta+h)/2<\varepsilon<(\Delta+h)/2$	$\dot{\varepsilon} > 0$	(B-23)

It has been shown that the quantity $q = Km_s - f$ is a function of both the force on the piston and the dry friction. Thus this quantity q

can be considered as the input of the linear element. However the force Km_s on the piston depends on the hysteresis and deadzone of the nonlinear element, while the dry friction f relies on the velocity of the output of the linear element. Equation (B10) gives the junction point between the nonlinear and linear elements.

The key equation in our analysis is equation (B9) where τ is given in equation (18) and q is shown in equations (B10) to (B23) for the nonlinear elements. This equation can be further simplified by multiplying through by τ and by using the following transformation:

$$P = K M\tau \tag{B24}$$

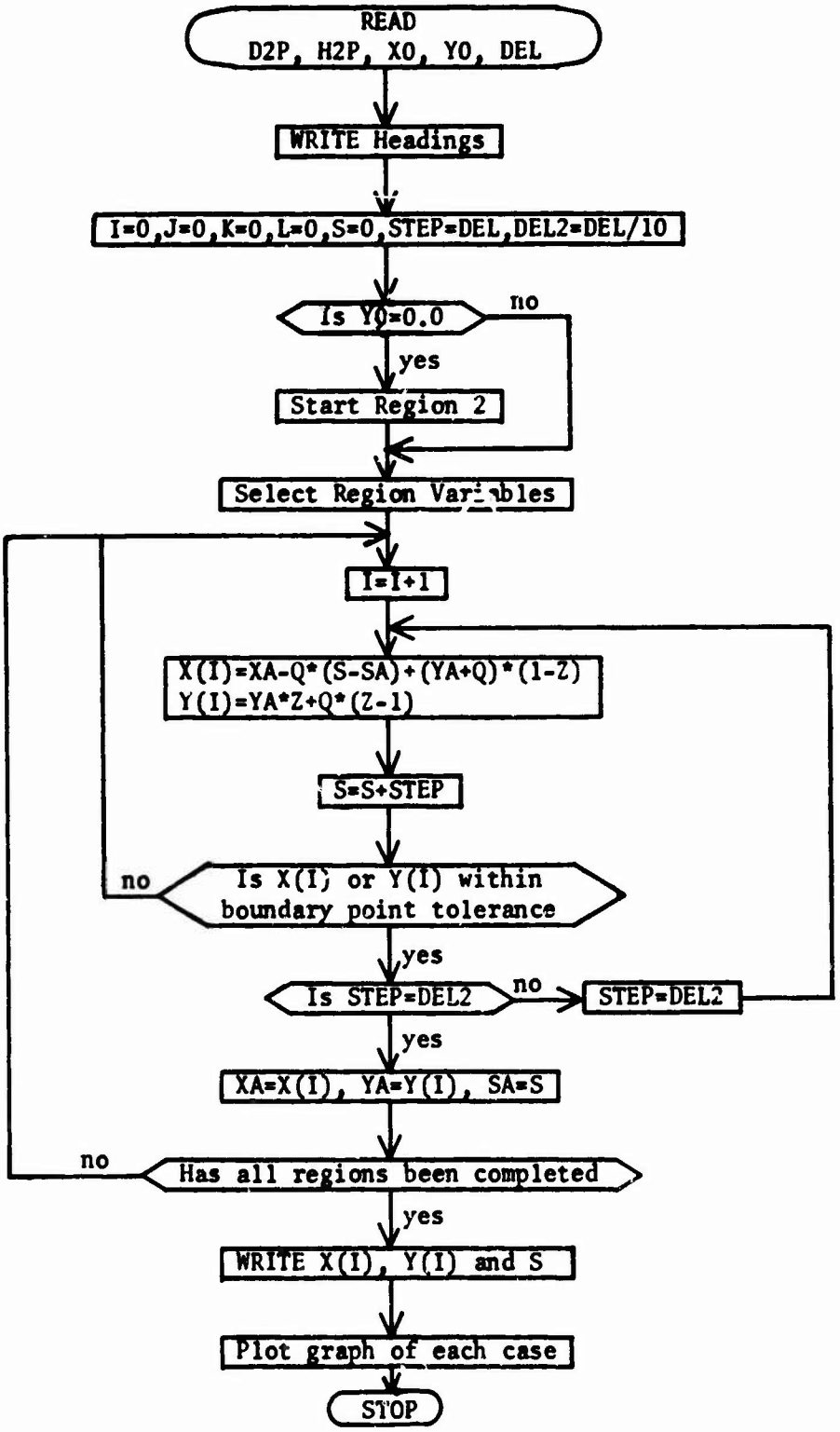
$$x = \frac{\varepsilon}{P} = \frac{\varepsilon}{KM\tau} \tag{B25}$$

$$\sigma = t/\tau \tag{B26}$$

The result is given by equations (1) - (3) in the text.

APPENDIX C

THE FLOW CHART AND PROGRAM



```
      DIMENSION IBUF (2000), X(1500), Y(1500)
      CALL PLOTS (IBUF, 2000, 6 )
    1 READ (5,23,END=22) D2P,H2P,XO,YO,DEL,FKM
      DEL2 = DEL/10.0
      X2 = D2P-H2P
      X3 =-(D2P+H2P)
      X5 = -D2P+H2P
      X6 = D2P+H2P
      STEP = DEL
      S=0.0
      I=0
      J=0
      K=0
      L=0
      WRITE (6,24) XO,YO,D2P,H2P,DEL,FKM
      WRITE (6,27)
    2 L=L+1
      GO TO (3,19,19,21),L
    3 IF (YO.EQ. 0.0) GO TO 18
    4 K=K+1
      GO TO (12,13,14,15,16,17,2),K
    5 J=J+1
      GO TO (6,10,11),J
    6 KCHECK=0.0
    7 I=I+1
    8 Z=EXP(SA-S)
      X(I)=XA-0*(S-SA)+(YA+0)*(1-Z)
      IF (K.EQ. 1 .OR. K .EQ. 4) GO TO 9
      IF (K .EQ. 2 .OR. K .EQ. 3) XC=X(I)
      IF (K .EQ. 5 .OR. K .EQ. 6) XB=X(I)
      IF (XB-XC .GT. 0.0) GO TO 5
    9 Y(I) = YA*Z+0*(Z-1)
      IF (K .EQ. 1 .AND. Y(I) .LE. 0.0) GO TO 5
      IF (K .EQ. 3 .AND. Y(I) .GE. -0.0005) GO TO 20
      IF (K .EQ. 4 .AND. Y(I) .GE. 0.0) GO TO 5
      IF (K .EQ. 6 .AND. Y(I) .LE. 0.0005) GO TO 20
      XSAVE=X(I)
      YSAVE = Y(I)
      KCHECK=1.0
      WRITE (6,28) I,S,X(I),Y(I)
      S = S + STEP
      IF (J .GT. 1) GO TO 8
      GO TO 7
   10 WRITE (6,25)
      S=S-STEP
      STEP = DEL2
      S = S + STEP
      KCHECK=0.0
      GO TO 8
   11 IF (KCHECK .EQ. 0.0) I=I-1
```

```
      X(I)=XSAVE
      Y(I)=YSAVE
      WRITE (6,29) I,S,X(I),Y(I)
      WRITE (6,26)
      S=S-STEP
      YA = YSAVE
      XA=XSAVE
      SA=S
      STEP = DEL
      S=S+STEP
      J=0
      GO TO 4
   12 O=1+FKM
      SA=0.0
      XA=XO
      YA=YO
      GO TO 5
   13 O=1-FKM
      XB=X2
      GO TO 5
   14 O=-FKM
      XB=X3
      GO TO 5
   15 O=-1-FKM
      GO TO 5
   16 O=-1+FKM
      XC=X5
      GO TO 5
   17 O=FKM
      XC=X6
      GO TO 5
   18 O=1-FKM
      SA=0.0
      XA=XO
      YA=YO
      XB=X2
      K=2
      GO TO 5
   19 O=1+FKM
      K=1
      GO TO 5
   20 WRITE (6,30)
   21 K1 = I + 1
      K2 = I + 2
      CALL PLOT (15.0,-30.0,-3)
      CALL PLOT (0.5, 0.5, -3)
      CALL SCALE (X,  6.0,  I, 1)
      CALL SCALE (Y, 8.0,  I, 1)
      YM = (0.0-Y(K1))/Y(K2)
      XM = (0.0-X(K1))/X(K2)
```

```
      IF (Y(K1) .GT. 0.0) YM=0.0
      IF (X(K1) .GT. 0.0) XM=0.0
      CALL AXIS (0.0,YM ,16H          X PLANE,-16, 6.0,0.0,X(K1),X(K2))
      CALL AXIS (XM ,0.0,16H          Y PLANE,16, 8.0,90.0,Y(K1),Y(K2))
      CALL LINE (X, Y, I, 1, 0, 0)
      GO TO 1
   22 CALL PLOT (12.0, 0.0, 999)
   23 FORMAT (6F10.4)
   24 FORMAT ('1',//,28X,'PHASE PLANE VALUES',3X,'XO=',F6.3,3X,'YO=',
     X F6.3,3X,'D2P=',F6.4,3X,'H2P=',F5.4,3X,'DEL=',F6.3,3X,'FKM=',F5.4)
   25 FORMAT (4X, 'START OF SMALL STEPS')
   26 FORMAT (4X, 'END OF SMALL STEPS')
   27 FORMAT (4X,'I',11X,'S',12X,'X(I)',10X,'Y(I)')
   28 FORMAT (2X,I4,4X,F10.3,4X,F10.4,4X,F10.4)
   29 FORMAT (2X,I4,4X,F10.3,4X,F10.4,'=XA',2X,F10.4,'=YA')
   30 FORMAT (4X,'Y(I) IS APPROXIMATELY EQUAL TO ZERO')
      CALL EXIT
```

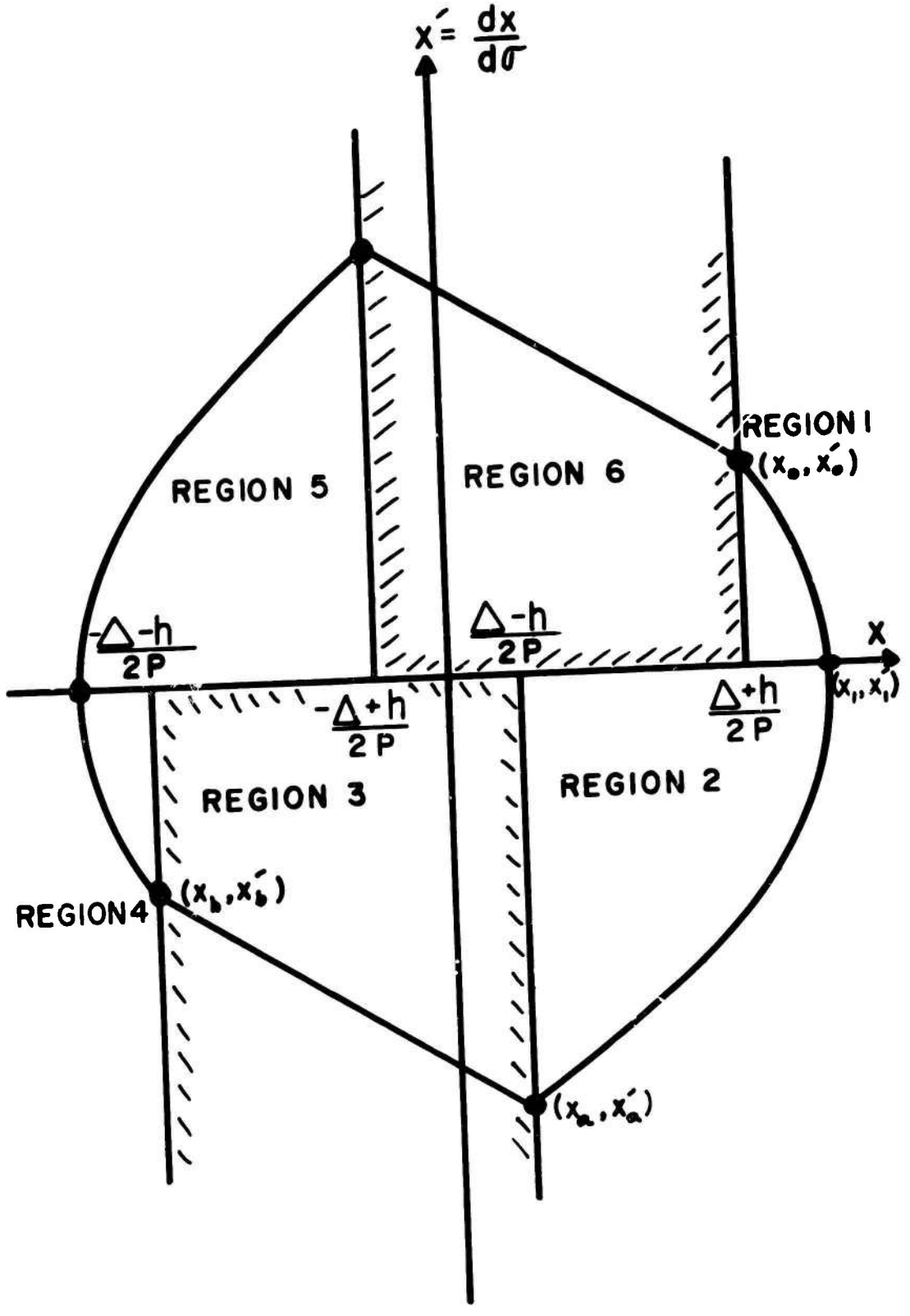

$x' = \frac{dx}{d\sigma}$
REGION 1
(x_o, x_o')
REGION 5
REGION 6
$\frac{-\Delta - h}{2P}$
$\frac{\Delta - h}{2P}$
X
(x_1, x_1')
$\frac{-\Delta + h}{2P}$
$\frac{\Delta + h}{2P}$
REGION 3
REGION 2
REGION4
(x_b, x_b')
(x_a, x_a')

FIGURE 1

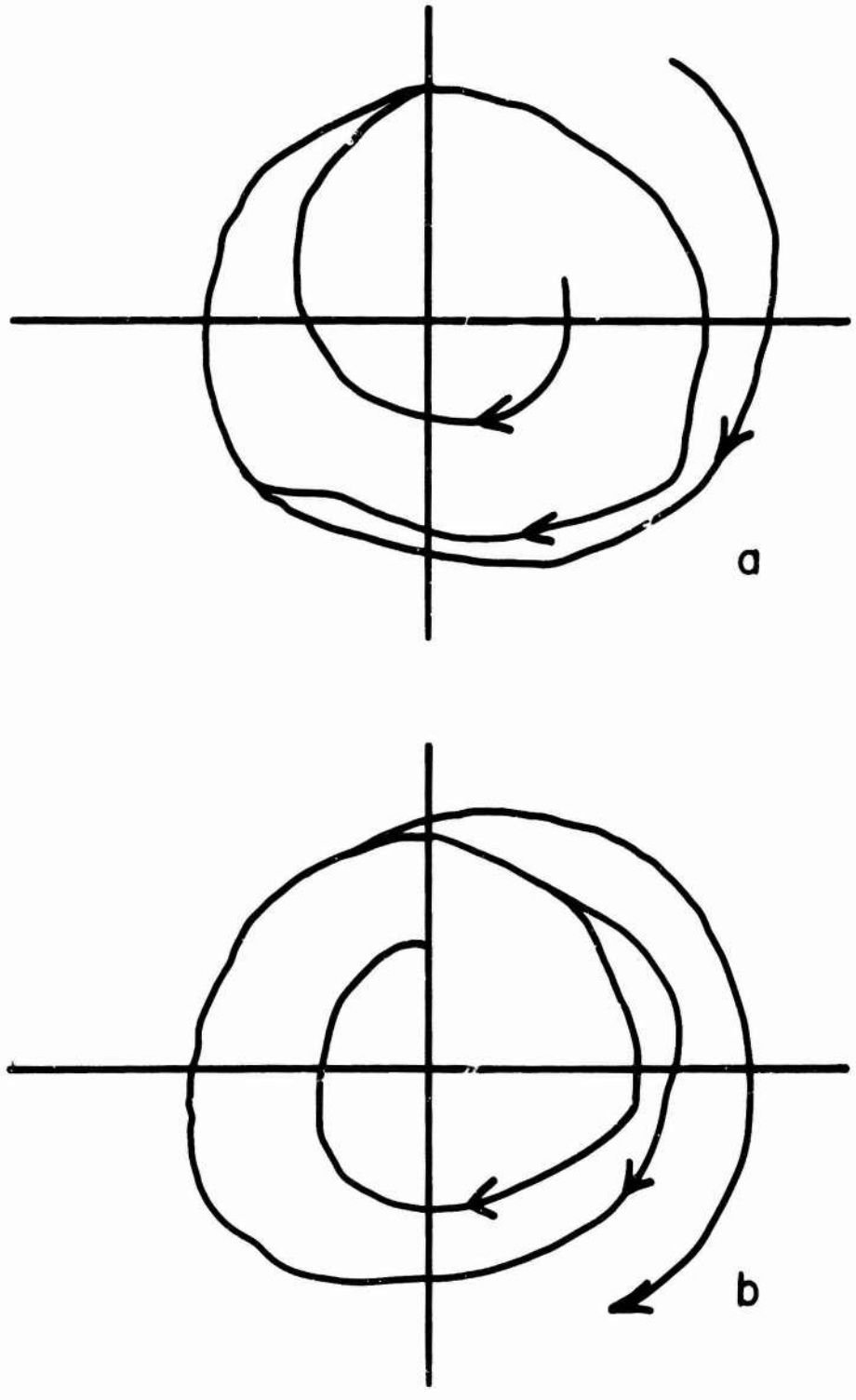


FIGURE 2

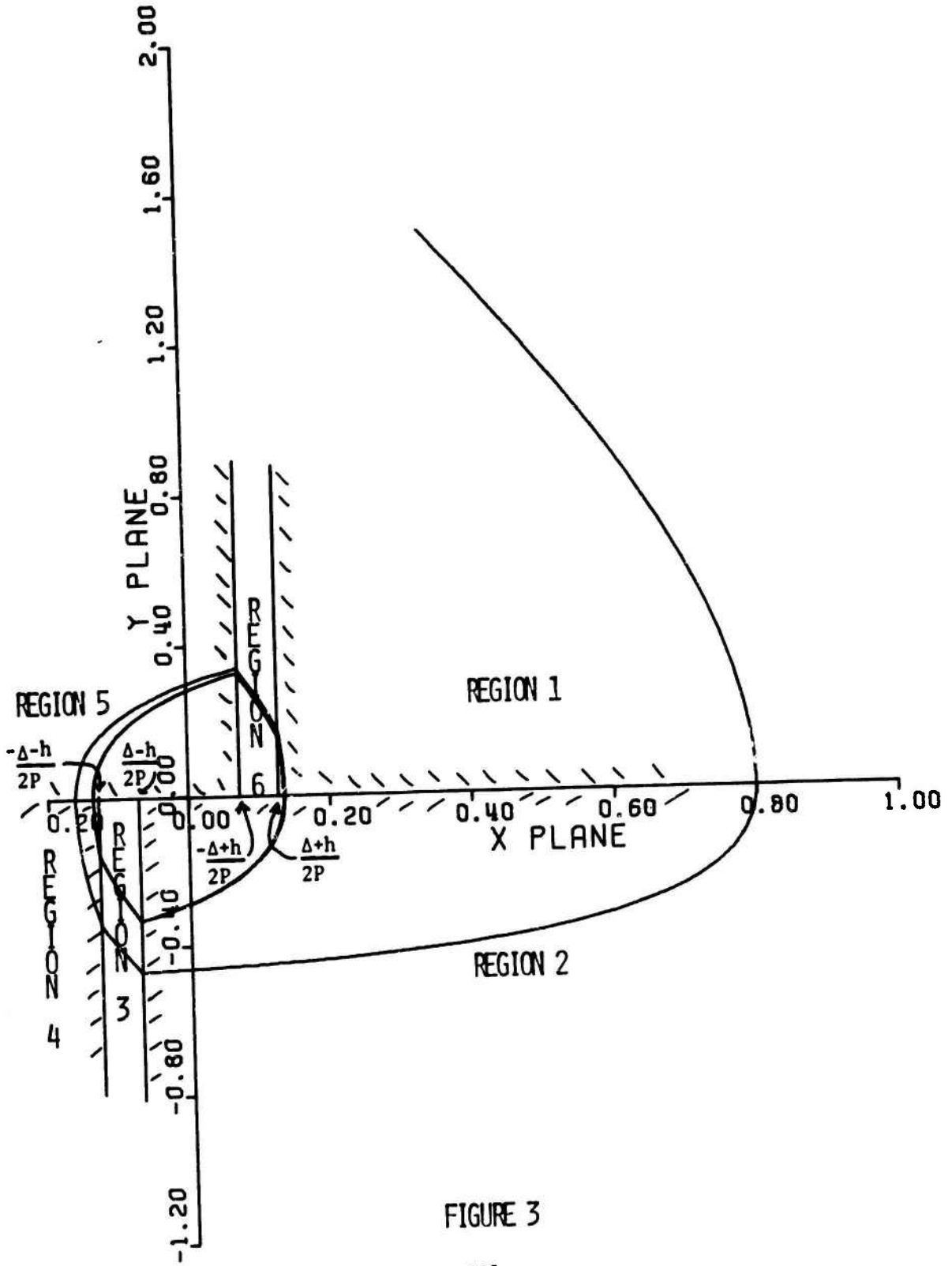

2.00
1.60
1.20
0.80
0.40
0.00
-0.40
-0.80
-1.20
Y PLANE
REGION 1
REGION 2
REGION 3
REGION 4
REGION 5
REGION 6
-Δ-h / 2P
Δ-h / 2P
-Δ+h / 2P
Δ+h / 2P
0.20
0.40
0.60
0.80
1.00
X PLANE

FIGURE 3

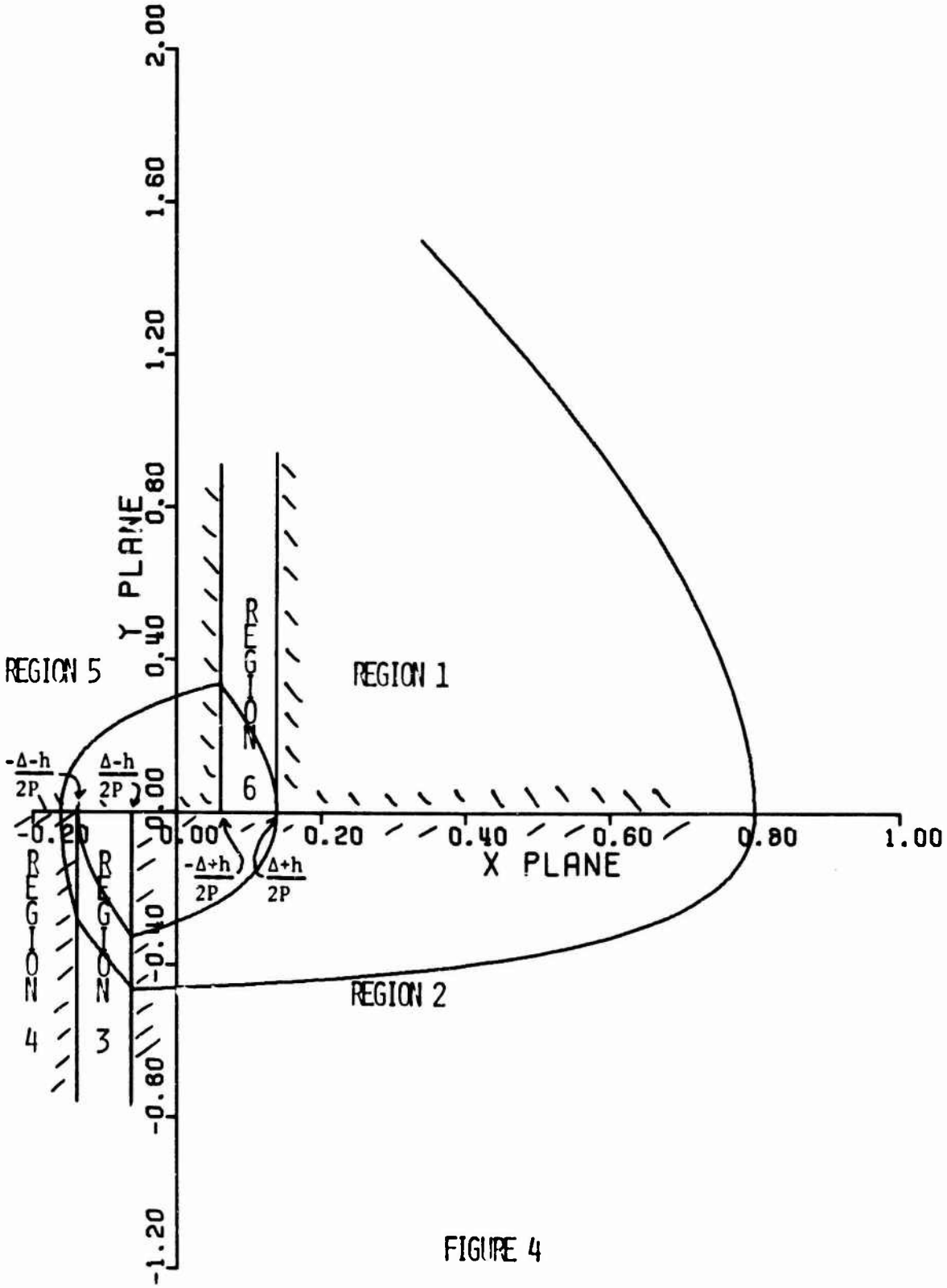

2.00
1.60
1.20
0.80
0.40
0.00
-0.40
-0.80
-1.20
Y PLANE
REGION 5
REGION 1
REGION 6
REGION 4
REGION 3
REGION 2
-Δ-h/2P
Δ-h/2P
-Δ+h/2P
Δ+h/2P
-0.20
0.00
0.20
0.40
0.60
0.80
1.00
X PLANE

FIGURE 4

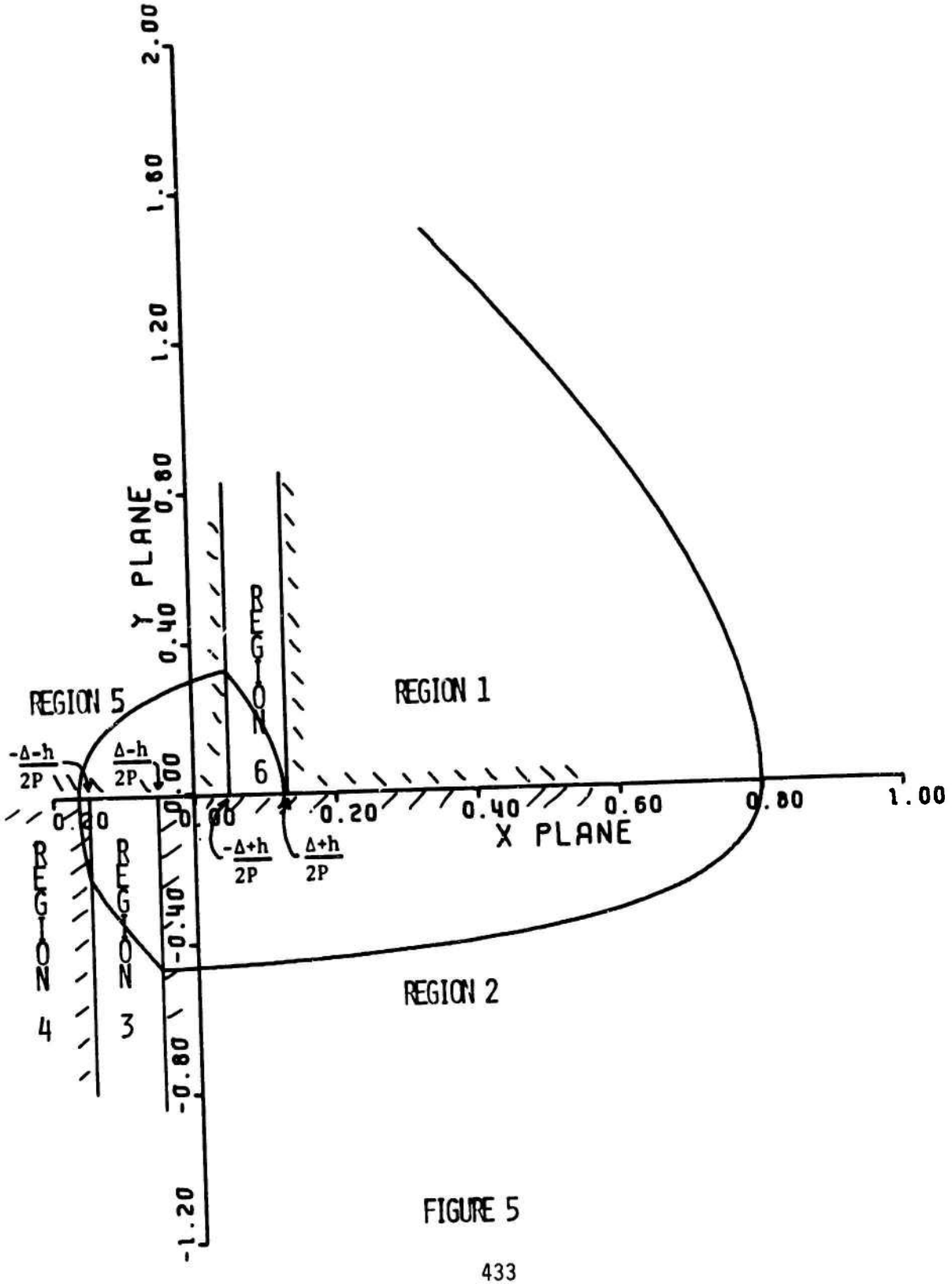

2.00
1.60
1.20
0.80
0.40
0.00
-0.40
-0.80
-1.20
Y PLANE
REGION 5
REGION 1
REGION 6
REGION 4
REGION 3
REGION 2
$\frac{-\Delta-h}{2P}$
$\frac{\Delta-h}{2P}$
$\frac{-\Delta+h}{2P}$
$\frac{\Delta+h}{2P}$
0.00
0.20
0.40
0.60
0.80
1.00
X PLANE

FIGURE 5

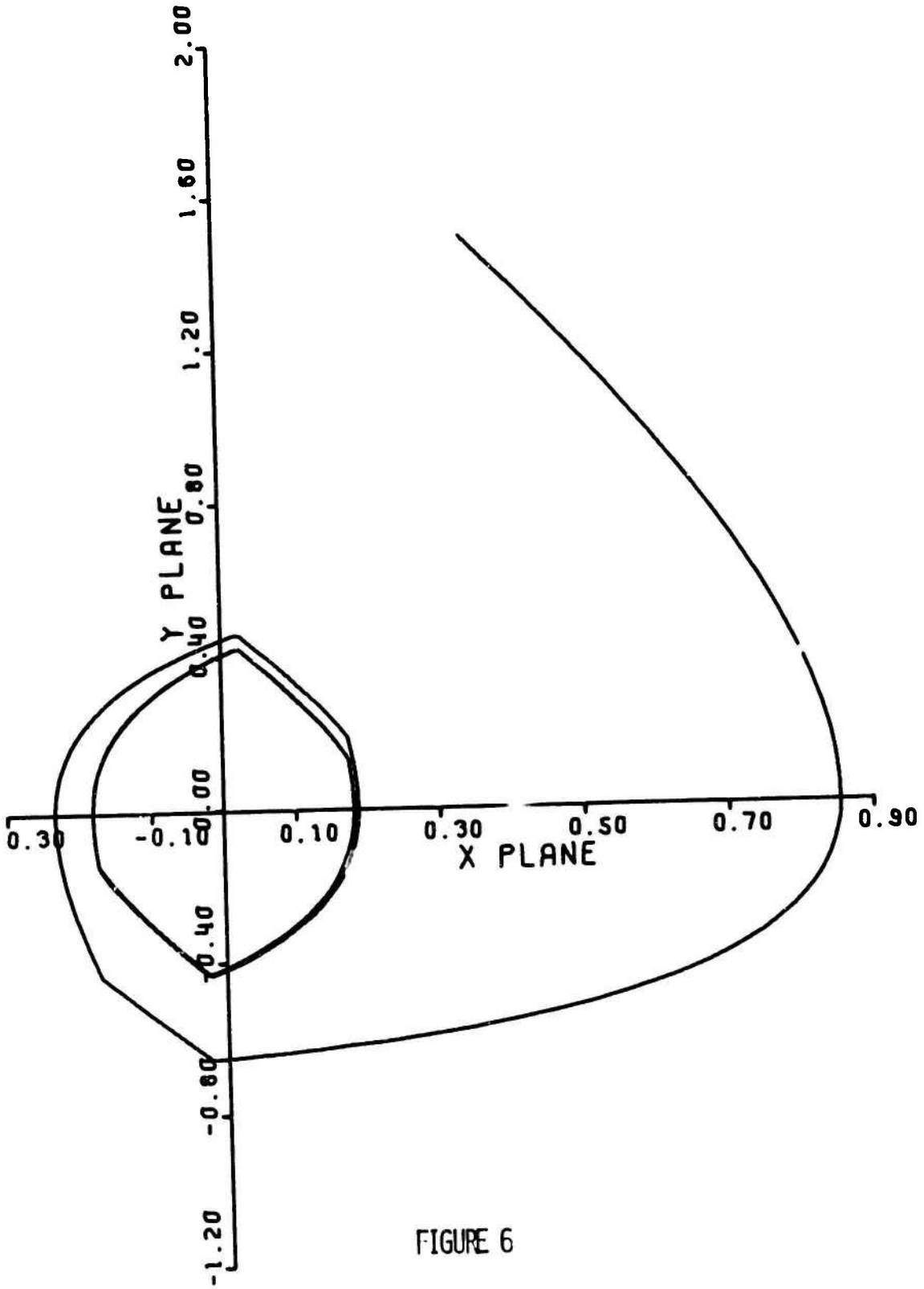

2.00
1.60
1.20
0.80
0.40
0.00
-0.40
-0.80
-1.20
Y PLANE
0.30
-0.10
0.10
0.30
0.50
0.70
0.90
X PLANE

FIGURE 6

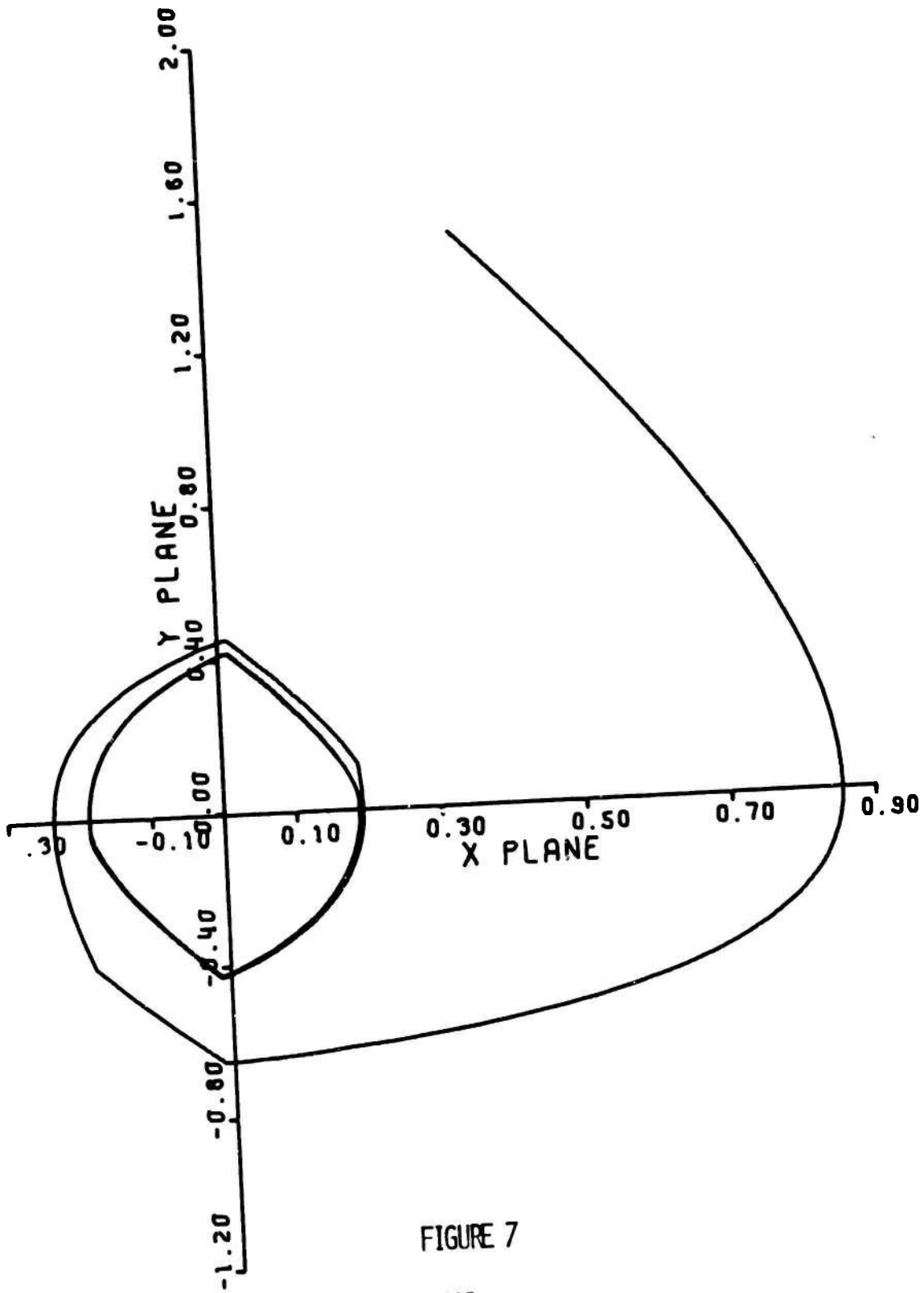

2.00
1.60
1.20
0.80
0.40
0.00
-0.40
-0.80
-1.20
Y PLANE
.30
-0.10
0.10
0.30
0.50
0.70
0.90
X PLANE

FIGURE 7

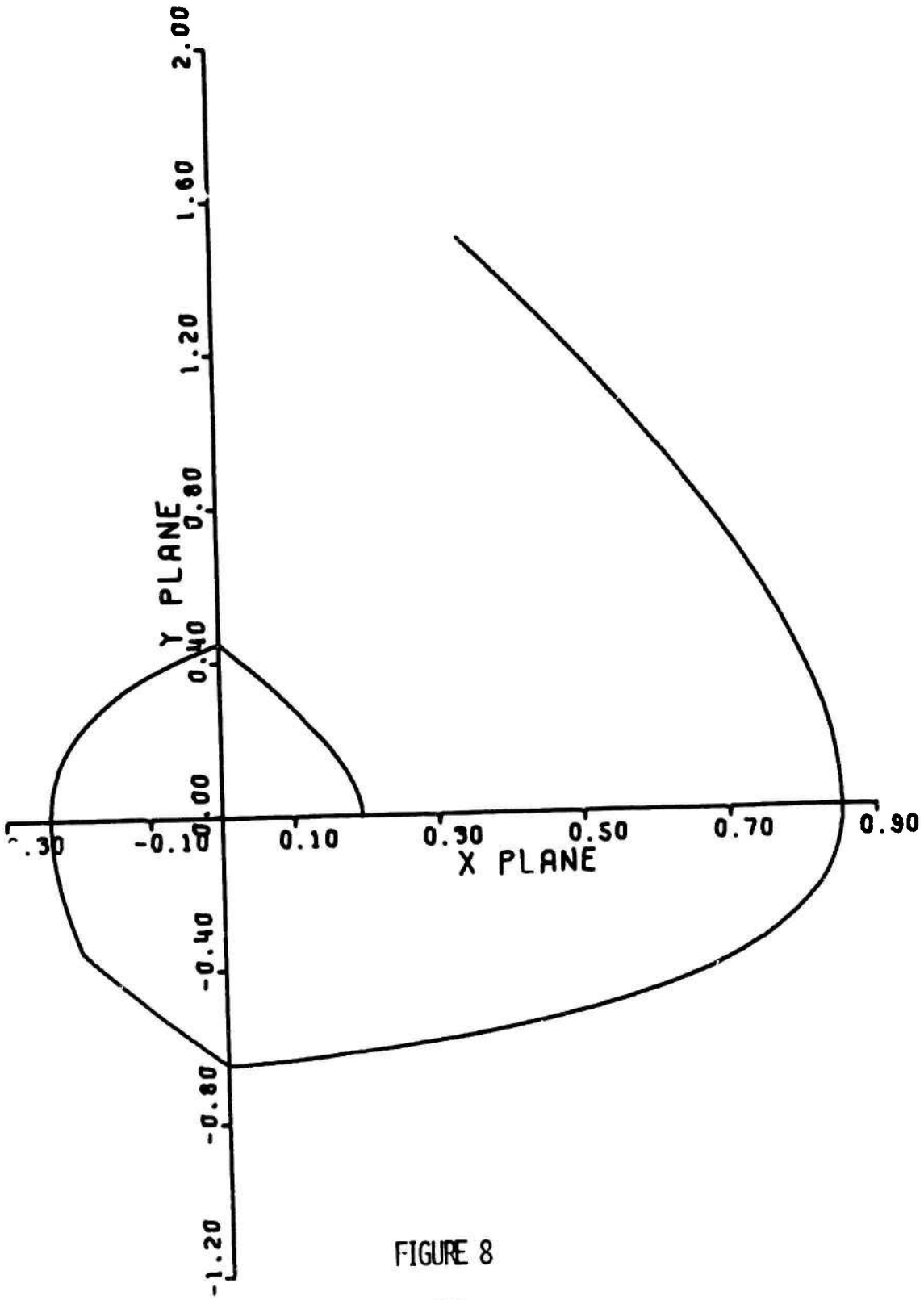

2.00
1.60
1.20
0.80
0.40
0.00
-0.40
-0.80
-1.20
Y PLANE
-0.30
-0.10
0.10
0.30
0.50
0.70
0.90
X PLANE

FIGURE 8

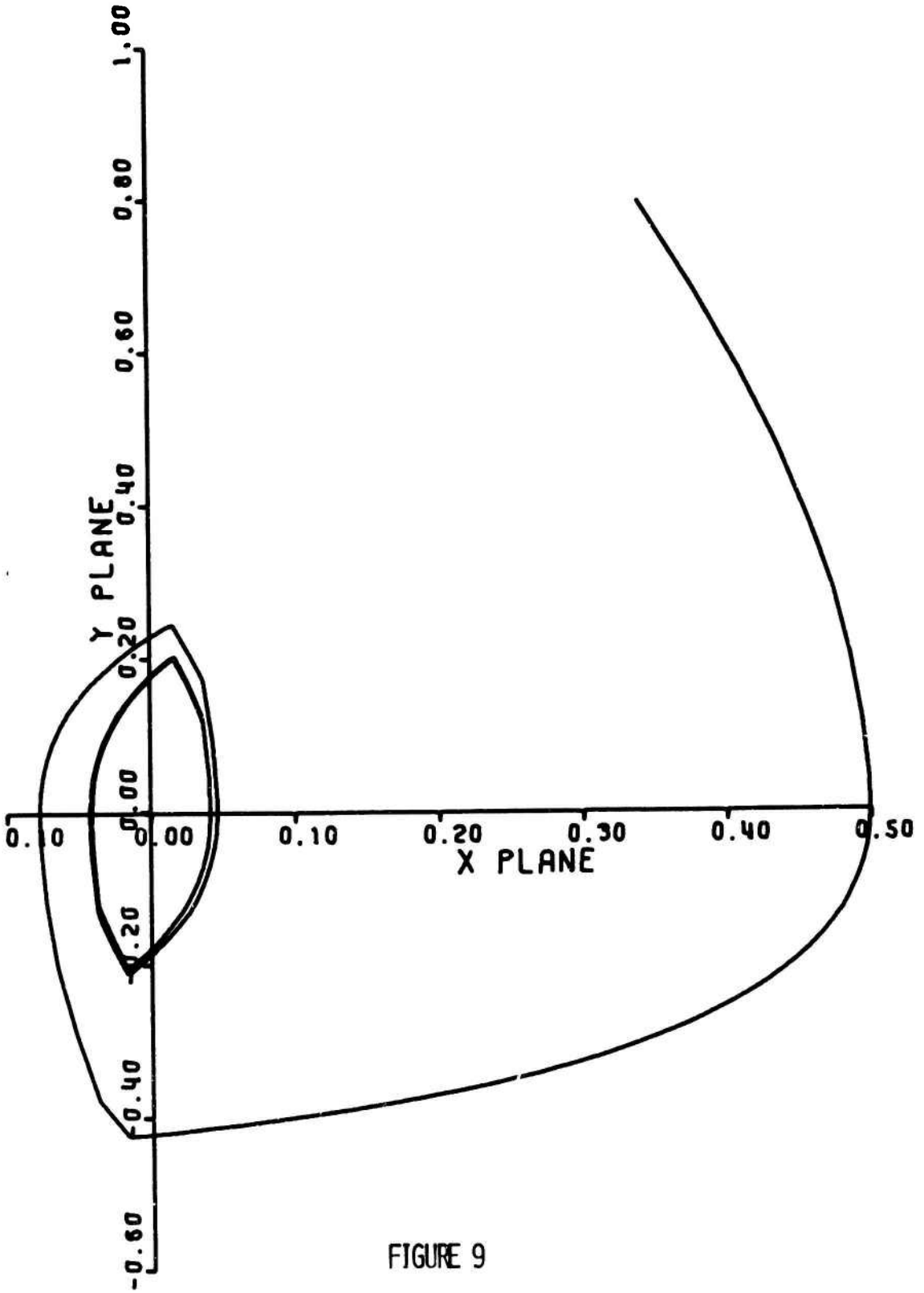

1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
Y PLANE
-0.10
0.00
0.10
0.20
0.30
0.40
0.50
X PLANE

FIGURE 9

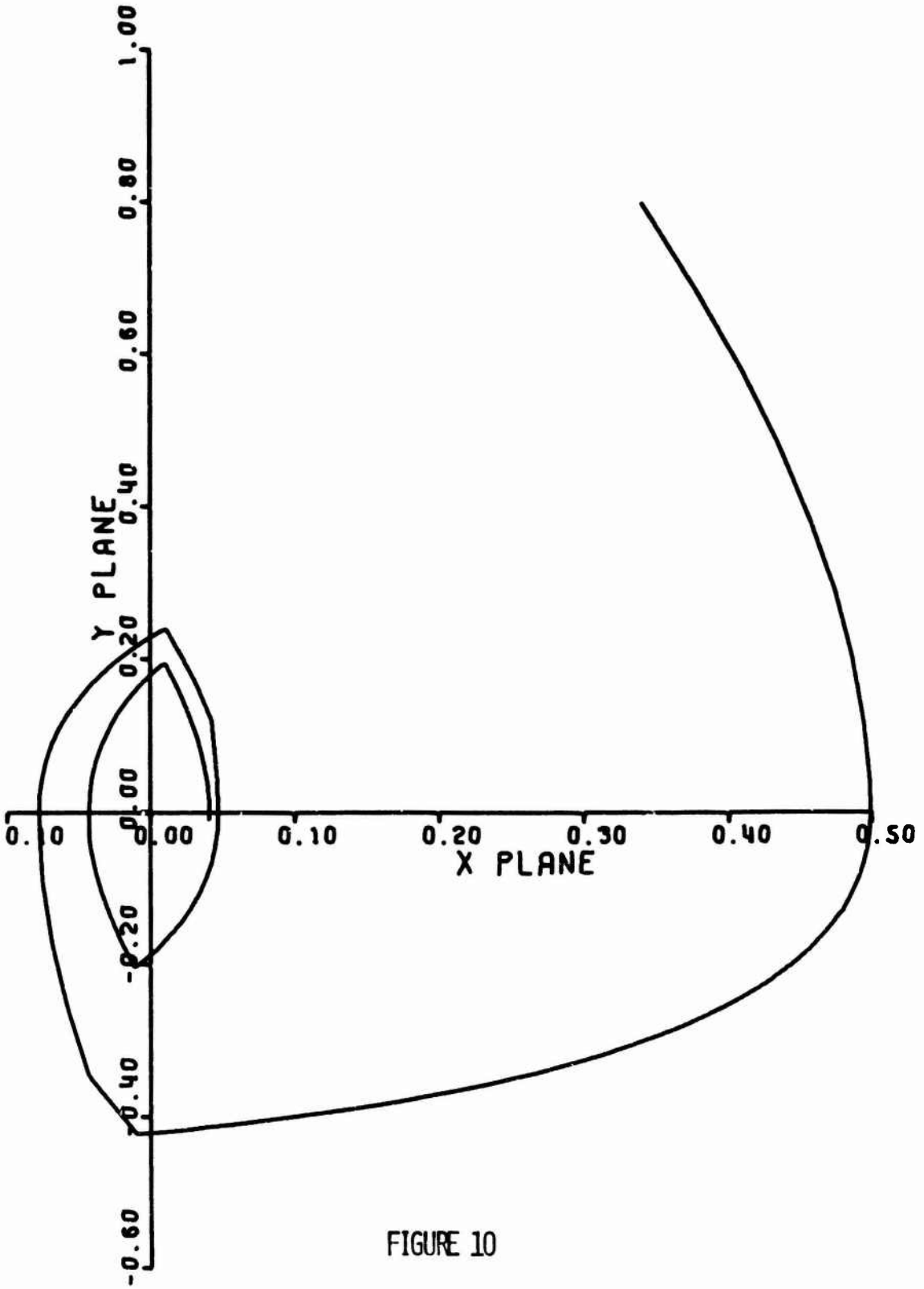

1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
Y PLANE
0.10
0.00
0.10
0.20
0.30
0.40
0.50
X PLANE

FIGURE 10

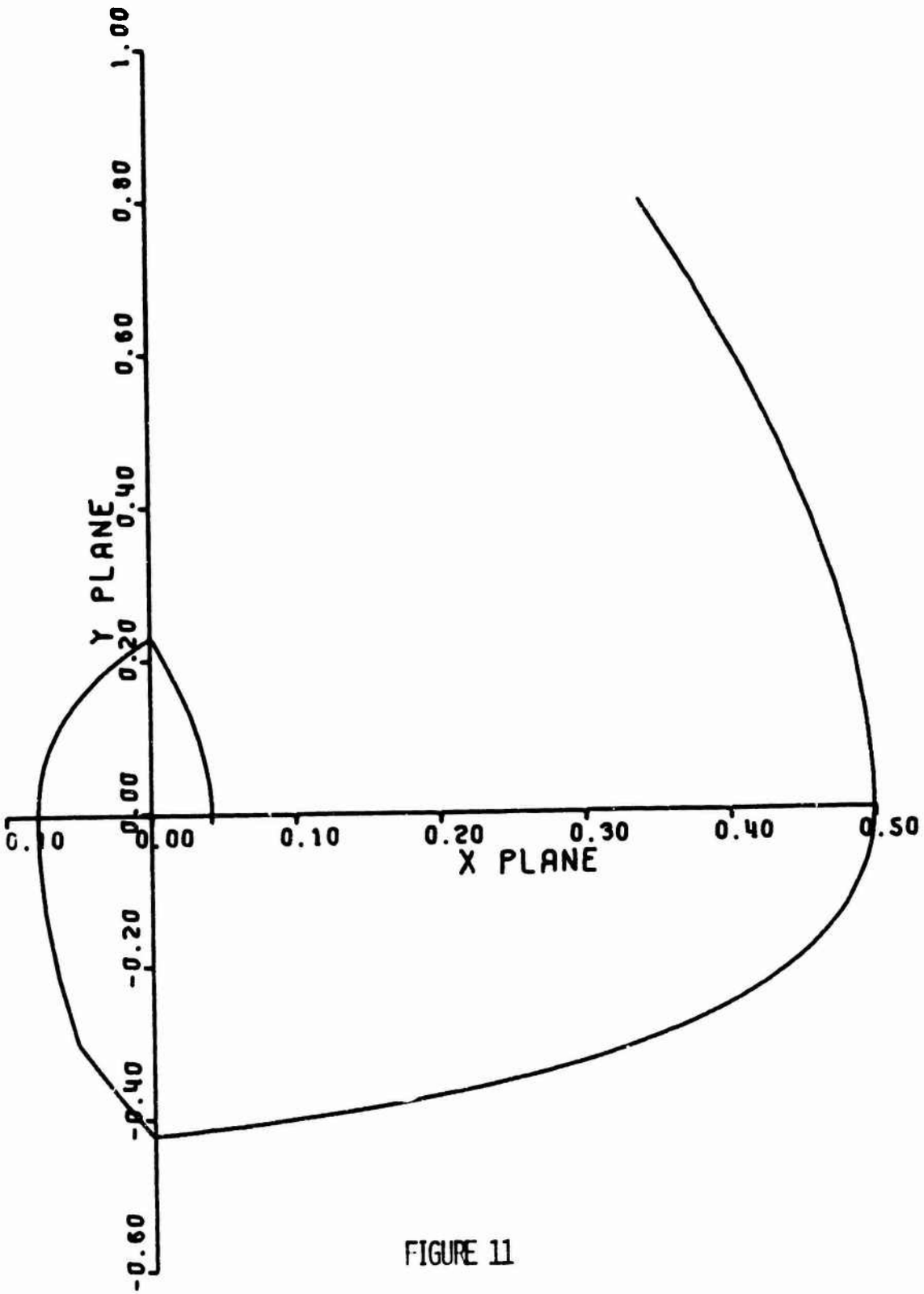

1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
Y PLANE
0.10
0.00
0.10
0.20
0.30
0.40
0.50
X PLANE

FIGURE 11

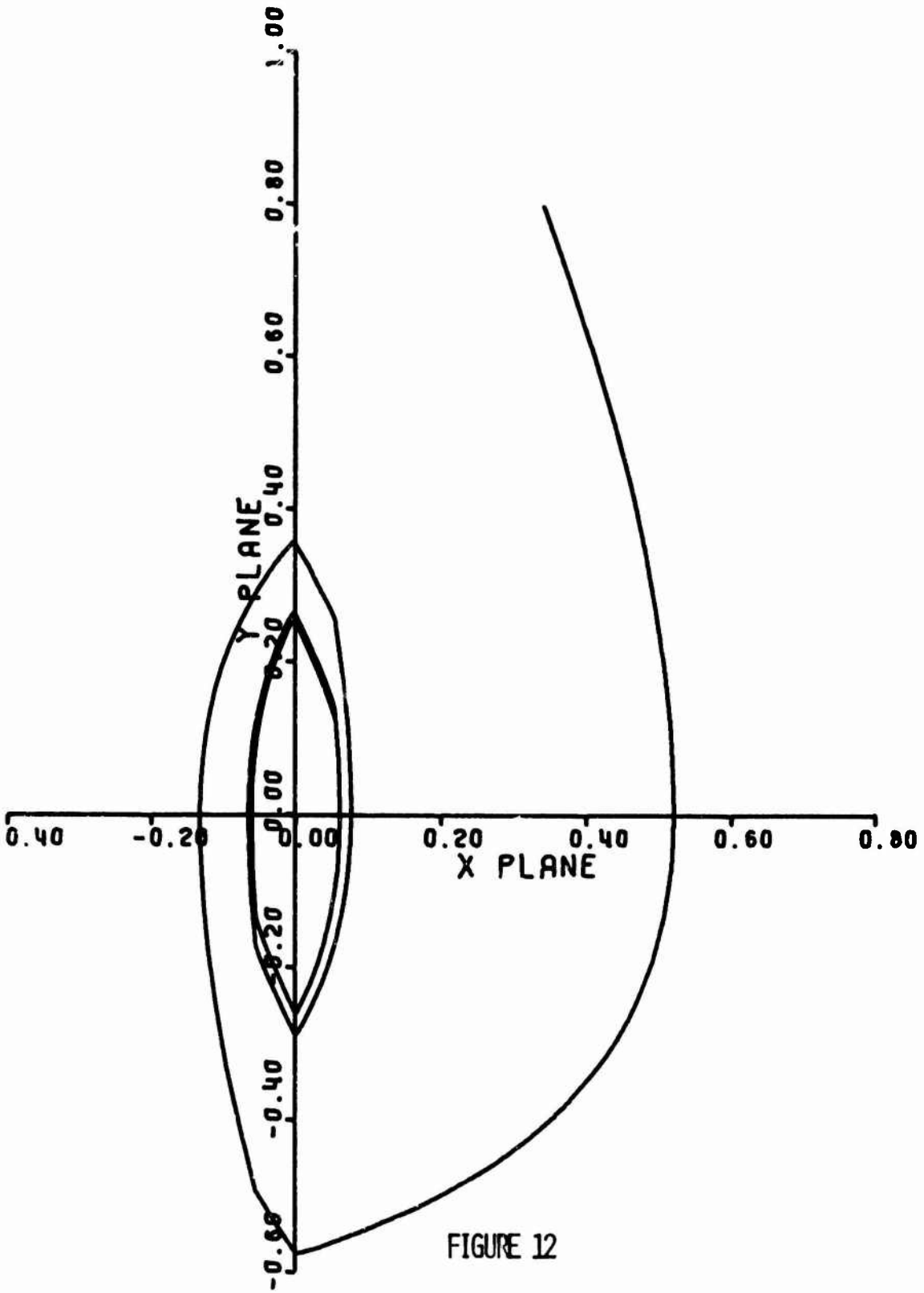

1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
Y PLANE
0.40
-0.20
0.00
0.20
0.40
0.60
0.80
X PLANE

FIGURE 12

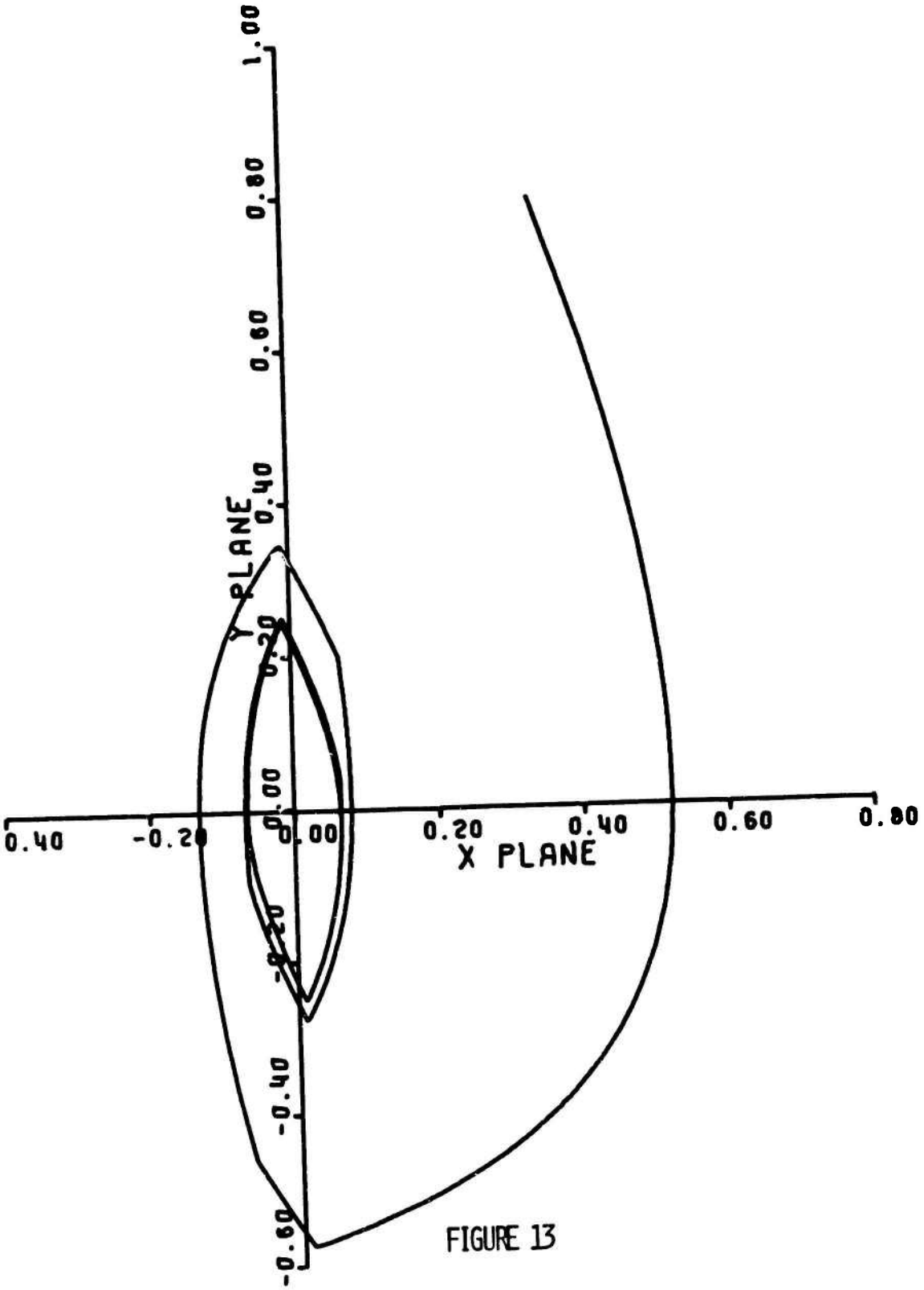

1.00
0.80
0.60
0.40
Y PLANE
0.20
0.00
-0.40
-0.60
0.40
-0.20
0.00
0.20
0.40
0.60
0.80
X PLANE

FIGURE 13

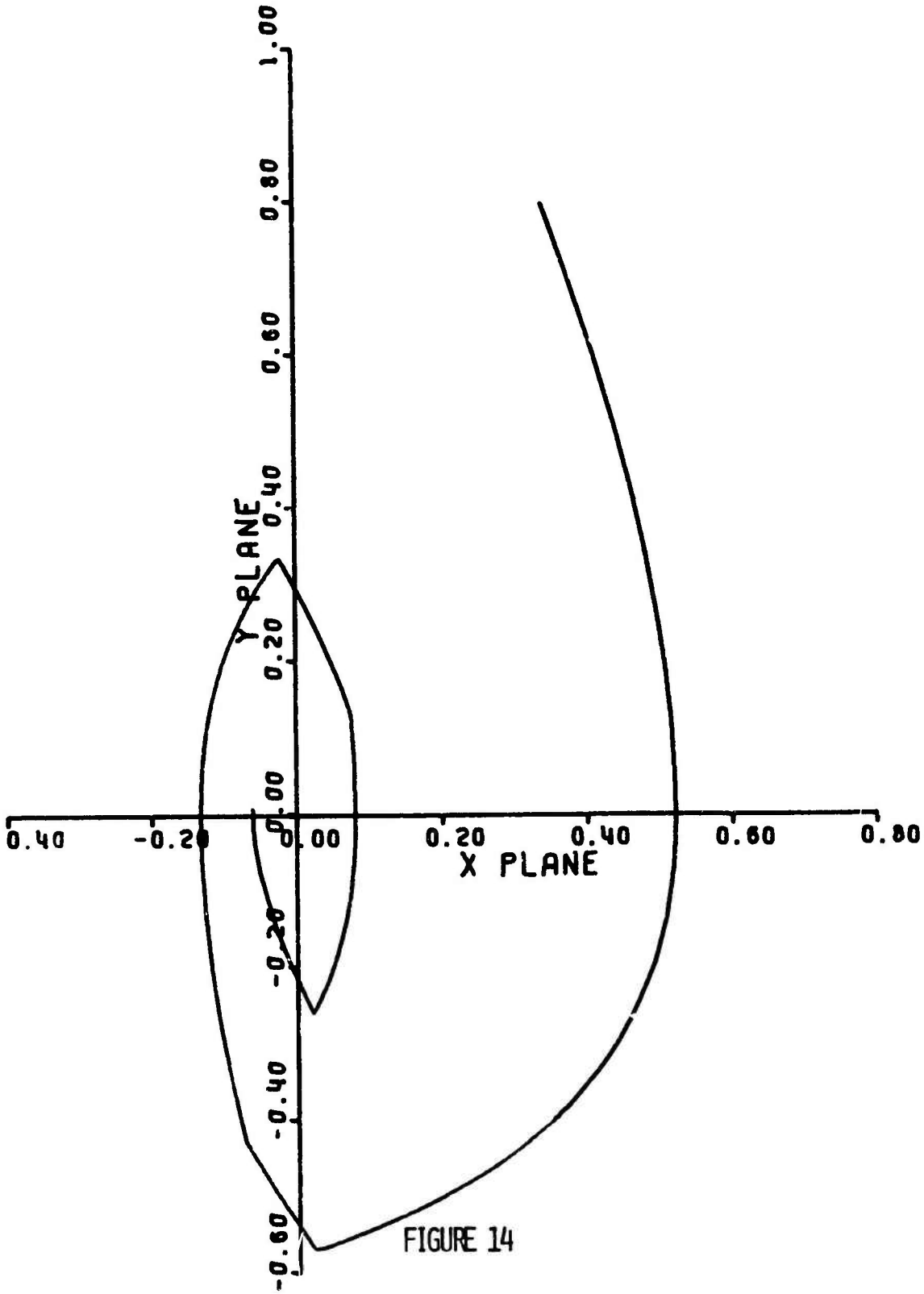

1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
Y PLANE
0.40
-0.20
0.00
0.20
0.40
0.60
0.80
X PLANE

FIGURE 14

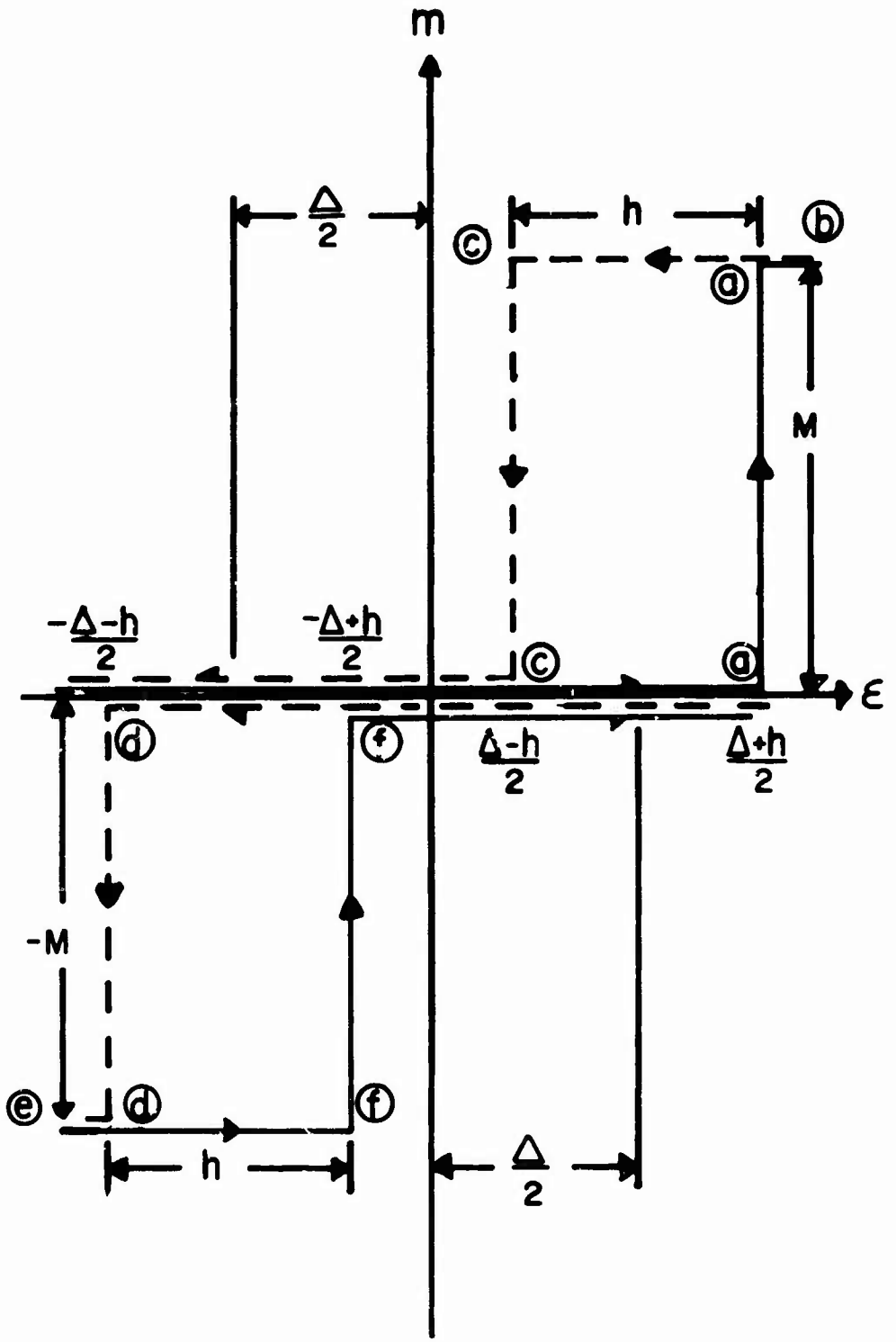


FIGURE A1

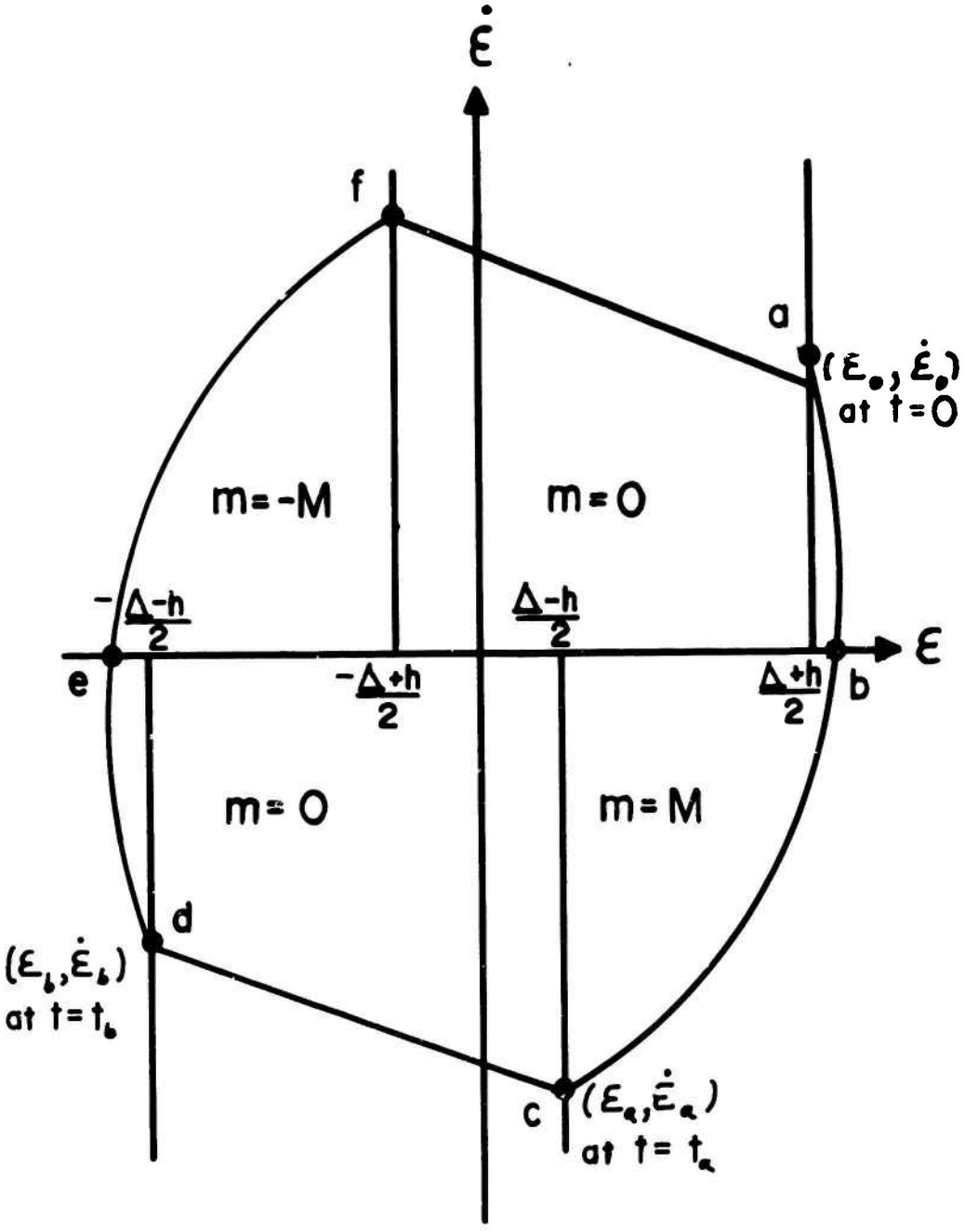

$\dot{\varepsilon}$
f
a
$(\varepsilon_a, \dot{\varepsilon}_a)$
at $t=0$
m=-M
m=0
$-\frac{\Delta-h}{2}$
$\frac{\Delta-h}{2}$
ε
e
$-\frac{\Delta+h}{2}$
$\frac{\Delta+h}{2}$
b
m=0
m=M
d
$(\varepsilon_b, \dot{\varepsilon}_b)$
at $t=t_b$
c
$(\varepsilon_c, \dot{\varepsilon}_c)$
at $t=t_c$

FIGURE A2

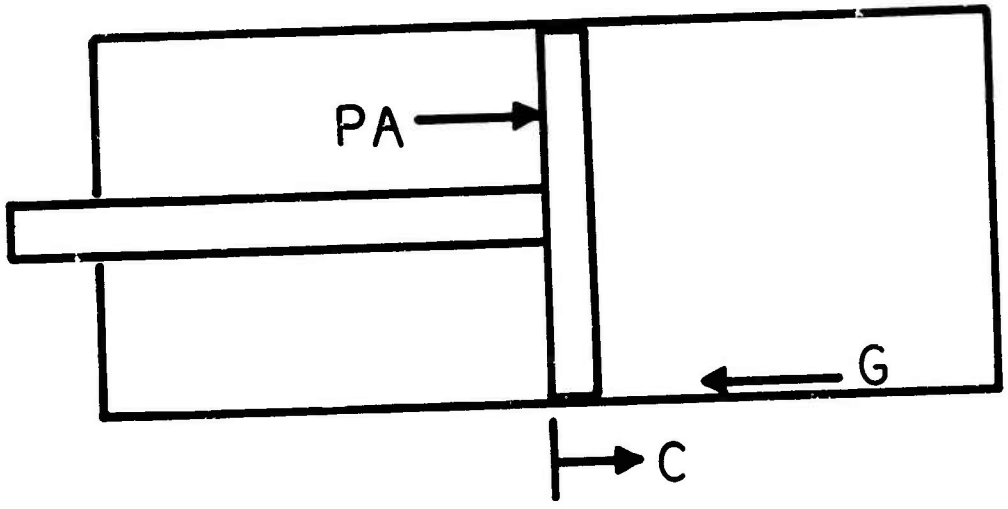


FIGURE B1

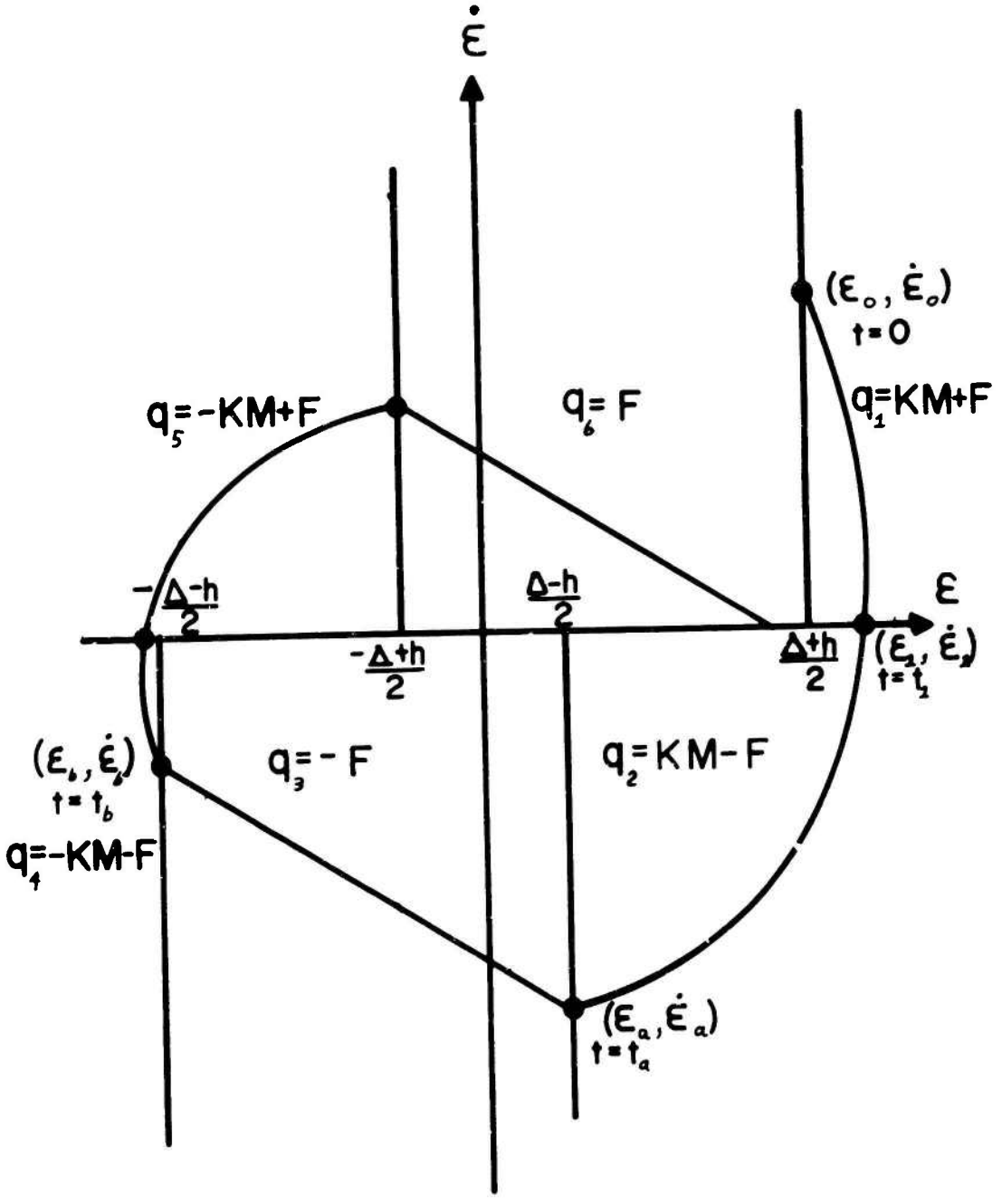

$\dot{\varepsilon}$
$(\varepsilon_o, \dot{\varepsilon}_o)$
$t=0$
$q_5=-KM+F$
$q_6=F$
$q_1=KM+F$
$-\frac{\Delta-h}{2}$
$\frac{\Delta-h}{2}$
ε
$-\frac{\Delta+h}{2}$
$\frac{\Delta+h}{2}$
$(\varepsilon_1, \dot{\varepsilon}_1)$
$t=t_1$
$(\varepsilon_b, \dot{\varepsilon}_b)$
$t=t_b$
$q_3=-F$
$q_2=KM-F$
$q_4=-KM-F$
$(\varepsilon_a, \dot{\varepsilon}_a)$
$t=t_a$

FIGURE B2

CONVERGENCE PROPERTIES OF QUASI-NEWTON METHODS WITH APPROXIMATE LINE SEARCHES

Melanie L. Lenard
Mathematics Research Center
University of Wisconsin-Madison
Madison, Wisconsin 53706

ABSTRACT. A typical iteration of a quasi-Newton method for unconstrained minimization consists of choosing a suitable direction and then searching for the minimum of the function along a line in that direction. Theoretical convergence properties of these methods usually require that the line search be exact, a condition which can never be satisfied in practice. Extensions of convergence theory to the case where approximate line searches are permitted will be presented, and computational experience with the approximate algorithms will be discussed.

Introduction

A number of recently developed techniques for finding the unconstrained minimum of $F(x)$, a differentiable function of several variables, may be described as quasi-Newton methods. These methods are iterative procedures, which, given a starting point x, search along a line from x in a direction d to a new point x^* with a decreased function value. The search direction is chosen to be

$$d = - Hg , \tag{1}$$

where $g = \nabla F(x)$ and H is a positive definite matrix which in some sense approximates the inverse of the Hessian, that is,

$$H \approx [\nabla^2 F(x)]^{-1} .$$

Typically a quasi-Newton method begins by setting H equal to the identity matrix on the first iteration and then improves the approximation to the inverse Hessian matrix by updating H at each subsequent iteration. The various methods differ from one another chiefly in the formula used for the updating process. Broyden [1967] has described a class of quasi-Newton methods where the following formula is used to calculate a new matrix H^*:

$$H^* = H - \frac{Hyy^T H}{y^T Hy} + \frac{ss^T}{s^T y} + \beta ww^T \tag{2}$$

Sponsored by the United States Army under Contract No. DA-31-124-ARO-D-462.

where $y = g^* - g = \nabla F(x^*) - \nabla F(x)$, $s = x^* - x$, and

$$w = \frac{s}{s^T y} - \frac{Hy}{y^T Hy} .$$

By appropriate choice of the parameter β one can obtain any of the important symmetric updating formulas. For example, $\beta = 0$ gives the Davidon [1959]- Fletcher-Powell [1963] formula. Also in this class is the rank-one correction formula [Broyden (1967), Murtagh and Sargent (1969)]

$$H^* = H + \frac{(s - Hy)(s - Hy)^T}{(s - Hy)^T s}$$

and the Broyden [1970]-Fletcher [1970]-Shanno [1969] formula:

$$H^* = H + \frac{ss^T}{s^T y}\left(1 + \frac{y^T Hy}{s^T y}\right) - \left(\frac{sy^T H + Hys^T}{s^T y}\right) .$$

Theoretical convergence properties of quasi-Newton methods usually depend on taking x^* to be such that

$$F(x^*) = \min\{F(x + td) \mid t > 0\} . \qquad (3)$$

Thus, a one-dimensional minimization problem must be solved at each iteration. Since this problem can consume a great deal of computational time, and since an "exact" solution can never be obtained, implementations often use approximate line searches.

In what follows, we will review what is known about convergence properties of quasi-Newton methods with exact line searches. We will then consider extensions of convergence theory to cover the practical situation where only an approximate solution to the problem (3) is obtained.

Convergence Properties with Exact Line Searches

One property shared by all quasi-Newton methods is that they will locate the minimum of a quadratic function of n variables in n iterations. This fact has led to the development of "restarted" versions [McCormick and Pearson (1969)] of quasi-Newton methods: at intervals of $n + 1$ iterations, the formula (2) is ignored and instead H is set equal to the identity matrix.

Before we discuss the behavior of quasi-Newton algorithms on non-quadratic functions, we need to define some terms.

If $\{x^k\}$ converges to ξ, then the order of convergence is the largest number p for which

$$\overline{\lim_{k \to \infty}} \frac{\|x^{k+1} - \xi\|}{\|x^k - \xi\|^p} \leq K \tag{4}$$

for some positive constant K.

If the order of convergence is one ($p = 1$), we usually say that the rate of convergence is "linear." If $p = 2$, we say that the rate of convergence is "quadratic." If $p > 1$, or if $K = 0$ when $p = 1$, we say that the rate of convergence is "superlinear."

Now, it was proved by Powell [1971] that the Davidon-Fletcher-Powell (D-F-P) method is superlinearly convergent, provided that $F(x)$ is twice continuously differentiable, strongly convex, and satisfies a Lipschitz condition on the second derivatives at the solution.

McCormick [1969] has shown that, under the same conditions on the objective function, the restarted version of the D-F-P method is n-step quadratic [that is, if each group of n-steps is considered as one step, then $p = 2$ in formula (4)].

Further, Dixon [1972] has shown that all formulas belonging to Broyden's family of quasi-Newton methods generate identical points when applied to the minimization of quite general functions. Thus, the convergence properties of the D-F-P method given above apply to all methods in the class described by equation (2).

Approximate Line Searches

Many techniques for dealing with approximate line searches have been proposed.

One of the oldest is subrelaxation, where the step-size is taken to be some fraction of the optimal step-size:

$$t = \rho t^*, \quad 0 < \rho \leq 1 .$$

Goldstein and Price [1962] suggested that the step-size t should satisfy

$$\epsilon < \frac{F(x + td) - F(x)}{t \nabla F^T(x) d} < 1 - \epsilon$$

where $0 < \epsilon < 1$.

Armijo [1966] proposed letting

$$t = \beta^j \bar{t}$$

where $\bar{t} > 0$ is an initial guess for the step-size and j is the smallest integer for which

$$\frac{F(x + td) - F(x)}{t \nabla F^T(x) d} < 1 - \epsilon' , \quad 0 < \epsilon' < \tfrac{1}{2} .$$

Wolfe [1969] discussed convergence properties with many types of step-sizes, including

$$\frac{\nabla F^T(x + td) d}{\nabla F^T(x) d} < 1 - \epsilon , \quad 0 < \epsilon < 1 . \tag{5}$$

In his book, Polak [1971] gives convergence and convergence rate proofs for a number of algorithm using step-size rules of the Goldstein-Price or Armijo type. Results for step-size rules of type (5) have been obtained for conjugate gradient methods [Lenard (1972a), (1972b)].

In the next section, we will describe convergence results for a quasi-Newton method using the inequality (5) to define a step-size.

D-F-P Method - Approximate Line Searches

We will let

$$\theta(t) = \frac{\nabla F^T(x + td) d}{\nabla F^T(x) d}$$

We note that an exact line search is described by

$$\theta(t) = 0 .$$

We will require throughout that

$$0 \leq \theta(t) \leq 1 - \eta , \qquad 0 < \eta < 1 . \tag{6}$$

It has been shown [Lenard (1973)] that if at each iteration, we require that the error satisfy the following conditions:

$$\theta \leq Sg^{T}Hg/\|g\|^{2} \tag{7}$$

$$\theta \leq Tg^{*T}Hg^{*}/\|g\|^{2} \tag{8}$$

$$\theta^{2} \leq (1-c)g^{*}Hg^{*}/gHg, \quad 0 < c < 1 \tag{9}$$

where S and T are arbitrary positive constants, then convergence is linear. The rate of convergence is superlinear if one replaces condition (9) with the more stringent condition

$$\theta \leq (1-c)g^{*}Hg^{*}/gHg, \quad 0 < c < 1. \tag{9'}$$

For the above results, the assumptions about the objective functions were the same as those used by Powell [1971] for the case of exact line searches.

For the restarted version of the D-F-P method, the rate of convergence is n-step quadratic provided that in each cycle of n-steps (beginning at x^{1} and terminating at x^{n+1}), we require, in addition to conditions (7), (8), and (9), that

$$\theta^{k} \leq K\|\nabla F(x^{1})\|, \qquad k = 1, 2, \ldots, n$$

where K is an arbitrary positive constant.

Thus, we have described conditions on $\theta(t)$ which are sufficient to establish the same order of convergence as that known to apply when line searches are exact.

Computational Experience

As a demonstration of the use of approximate line searches, such as those described in the previous section, some computational experiments were performed. The results will be reported in this section.

The Davidon-Fletcher-Powell method was used to find the minimum of three classic test problems: Rosenbrock's [1960] banana-shaped

valley, the helical valley [Fletcher and Powell (1963)], and a 5-dimensional trigonometric problem [Fletcher and Powell (1963)]. We considered a problem solved if $\|\nabla F(x)\| \leq 10^{-4}$.

The one-dimensional minimization problem was solved by a method which uses function values only. The technique begins by finding three points bracketing the minimum and then uses quadratic interpolation to get successively better estimates of the minimum. This is continued until the value of the directional derivative (estimated by differences in function values) at the approximate minimum meets the chosen error criteria.

As the error tolerances were reduced, we expected an increase in the number of function evaluations per iteration. However, we wished to investigate the effect of varying the error tolerances on the total amount of computational effort required to solve a problem. Therefore, we have recorded the total number of function evaluations and the number of gradient evaluations required to solve each problem. (The number of iterations is one less than the number of gradient evaluations.) It should be noted that one gradient evaluation is usually considered to be the equivalent of n function evaluations.

In the experiments we performed, we tested condition (6) by letting η take on values of 0.99, 0.9, 0.5, and 0.1. Further, conditions (7) and (8) were tested with S taking on values of 10, 100, 1000, and 10^6, and $T = 0.9\,S$. Finally, we used condition (9) with $c = 0.1$.

The results of these experiments are displayed in Table I (D-F-P with restart) and Table II (D-F-P continued). Although there is no obvious pattern, we can make several observations based on this data.

First, the number of iterations did not change dramatically with changes in error criteria. Second, the value of S seems to have little effect. Also, we note that checking conditions (8) and (9) requires a gradient evaluation at the end of the line search. If they are satisfied, calculations proceed in the normal way. If they are not satisfied, the solution to the line search must be refined and the gradient re-evaluated. In the few cases where this occurred, these "extra" gradient evaluations are indicated in parentheses in the tables.

Finally, it is interesting to see that D-F-P continued (super-linear convergence) performs better than D-F-P restarted (n-step quadratic convergence) on these test problems.

The major result, however, is that the error tolerance in the line search does not seem to be of great importance in the overall performance of the algorithm. Based on the data reported here, we expect that setting $\eta = 0.9$ in condition (6) and ignoring the other conditions would give satisfactory results.

Summary

In this paper, we have reviewed existing convergence theory for quasi-Newton methods, most of which is dependent on exact line searches. We have extended the theory to cover the practical situation where line searches are not exact. In particular, we have stated conditions on line search error under which the Davidon-Fletcher-Powell method has the usual order of convergence. Further, we have reported some computational experience which shows that the D-F-P method retains good performance when approximate line searches are permitted.

Table I

Davidon-Fletcher-Powell Method - Restarted Version

Problem	η	$S = 10^6$ #F	$S = 10^6$ #G	$S = 10$ #F	$S = 10$ #G	$S = 10^2$ #F	$S = 10^2$ #G	$S = 10^3$ #F	$S = 10^3$ #G
	0.99	245	28	308	28	245	28	252	30
Banana	0.9	229	30	235	30	366	28	299	31
Valley	0.5	223	31	229	29	244	29	276	30
n = 2	0.1	290	34	229	29	244	29	422	30
	0.99	346	29	346	29	346	29	254	27
Helical	0.9	207	29	208	29	231	28	221	28
Valley	0.5	206	30	224	29	228	28	219	28
n = 3	0.1	243	36	224	29	228	28	219	28
	0.99	181	12	115	12	110	13	95	12
Trig	0.9	83	12	91	12	106	13	103	12
Equations	0.5	103	14	90	12	106	13	103	12
n = 5	0.1	125	19	90	12	106	13	103	12

#F = Number of function evaluations.

#G = Number of gradient evaluations.

Table II

Davidon-Fletcher-Powell Method - Continued Version

Problem	η	$S = 10^6$ #F	$S = 10^6$ #G	$S = 10$ #F	$S = 10$ #G	$S = 10^2$ #F	$S = 10^2$ #G	$S = 10^3$ #F	$S = 10^3$ #G
	0.99	204	21	207	22	204	21	208	24
Banana	0.9	167	22	182	19	176	23	155	20
Valley	0.5	136	21	195	22	163	21	178	24(+1)*
n = 2	0.1	154	25	195	22	163	21	167	20(+3)*
	0.99	171	20	170	20	171	20	194	21
Helical	0.9	177	24	164	21	131	17	142	19
Valley	0.5	183	26	288	24	136	18	169	25
n = 3	0.1	188	27	288	24	136	18	190	26(+2)*
	0.99	84	10	83	10	90	10(+2)*	91	10
Trig	0.9	68	10	69	10	90	10(+1)*	86	10
Equations	0.5	69	11	69	10	90	10(+1)*	86	10
n = 5	0.1	67	11	69	10	90	10(+1)*	86	10

*Numbers in parentheses indicate additional gradient evaluations required to satisfy conditions (8) and (9).

#F = Number of function evaluations.

#G = Number of gradient evaluations.

REFERENCES

1. Armijo, L. (1966), "Minimization of functions having continuous partial derivatives", Pacific J. Math. 16, 1-3.

2. Broyden, C. G. (1967), "Quasi-Newton methods and their application to function minimization", Maths. of Comp. 21, 368-381.

3. Broyden, C. G. (1970), "The convergence of a class of double-rank minimization algorithms 2", J. Institute of Maths. and Applications 6, 222-231.

4. Davidon, W. C. (1959), "Variable metric method for minimization", A.E.C. Research and Development Report ANL-5990.

5. Dixon, L. C. W. (1972), "Quasi-Newton algorithms generate identical points", Math. Programming 2, 383-387.

6. Fletcher, R. (1970), "A new approach to variable metric methods", Computer J. 13, 317-322.

7. Fletcher, R. and Powell, M. J. D. (1963), "A rapidly convergent descent method for minimization", Computer J. 6, 163-168.

8. Goldstein, A. A. and Price, J. F. (1962), "An effective algorithm for minimization", Numerische Mathematik 10, 184-189.

9. Lenard, M. L. (1972a), "Practical convergence conditions for unconstrained optimization", Math. Programming 4, 309-323.

10. Lenard, M. L. (1972b), "Practical convergence conditions for restarted conjugate gradient methods", MRC Technical Summary Report No. 1373, University of Wisconsin.

11. Lenard, M. L. (1973), "Practical convergence conditions for the Davidon-Fletcher-Powell method", MRC Technical Summary Report No. 1356, University of Wisconsin.

12. McCormick, G. P. (1969), "The rate of convergence of the reset Davidon variable metric method", MRC Technical Summary Report No. 1012, University of Wisconsin.

13. McCormick, G. P. and Pearson, J. D. (1969), "Variable metric methods and unconstrained optimization", in Optimization, ed. R. Fletcher, Academic Press (London), 307-325.

14. Murtagh, B. A. and Sargent, R. W. H. (1969), "A constrained minimization method with quadratic convergence", in Optimization, ed. R. Fletcher, Academic Press (London), 215-246.

15. Polak, E. (1971), Computational Methods in Optimization: a Unified Approach, Academic Press, New York.

16. Powell, M. J. D. (1971), "On the convergence of the variable metric algorithm", J. Institute of Maths. and Applications 7, 21-36.

17. Rosenbrock, H. H. (1960), "An automatic method for finding the greatest or least value of a function", Computer Journal 3, 175-184.

18. Shanno, D. F. (1969), "Conditioning of quasi-Newton methods for function minimization", Center for Mathematical Studies Report No. 6910, University of Chicago.

19. Wolfe, P. (1969), "Convergence conditions for ascent methods", S.I.A.M. Review 11, 226-235.

ADAPTIVE NONLINEAR ESTIMATION APPLICATION FOR TEMPERATURE FORECASTING

LTC N. B. PENROSE

Department of Electrical Engineering
U. S. Military Academy
West Point, New York

ABSTRACT

This work is an application of adaptive estimation theory to temperature forecasting. It is presented as a feasibility study demonstrating the efficacy of the adaptive approach. The local station temperature forecasting problem is chosen to focus the discussion on the efficacy of the filtering algorithm by using only surface level, single geographic location data. The feasibility of the adaptive approach is established by comparison with the persistence forecast and other statistical methods.

While this paper is specific in discussing an adaptive estimation technique to predict the temperature process, the method and techniques are of general interest since they may be applied with equal ease to the prediction of any Gauss-Markov process.

I. Introduction

1.1 The Problem and General Approach to Solution

This paper discusses the application of adaptive nonlinear estimation theory to temperature forecasting. The techniques and results are intended to establish the feasibility of the method applied in order to form a basis upon which subsequent work may be extended. For this reason the local station, surface level temperature forecasting problem is chosen, i.e., only surface level, single station dry bulb temperature data is utilized for prediction purposes. It is recognized from the outset that correlating information of distant weather stations as well as atmospheric and other weather data are necessary to provide a statistically well supported temperature forecast. However, the method employed may be extended to include this additional information as required.

The approach taken is described as phenomenalistic: i.e., to drive the temperature forecasting model from the surface data available from a single geographic location. The techniques of adaptive nonlinear estimation are utilized to choose the particular functional form of the model to contain a finite set of parameters. These parameters are then learned, or optimized, such that the model best fits the measured output data (Lainiotis, 1971a).

To perform the model identification process, a training set of temperature data is u ilized. The relatively small size of the training set demonstrates the economic efficacy of the adaptive filtering approach. The class of nonlinear prediction problems where explicit solutions are available is that in which the system dynamical equations are specified within a finite set of unknown constant parameters in the form of linear differential or difference equations. One approach to this problem utilizes Bayes' rule (Bucy, 1965) as extended by Magill (1965) and Lainiotis (1971b) in the form of the Partition Theorem which asserts that the optimal estimation of the system states can be "partitioned" into a linear non-adaptive part consisting of ordinary Kalman filters matched to each admissible value of a parameter, θ, and a nonlinear part, consisting of likelihood ratios, that incorporates the adaptive, learning or system-identifying nature of the estimator. The application of this theorem results in the Adaptive Kalman filter which is depicted in Figure 1.

1.2 The Data

The temperature process results from a random nonlinear, periodic nonstationary system (Jones, 1971). The temperature process is assumed statistically to be sufficiently Gauss-Markov to satisfy the requirements of the Kalman filter within reasonable bounds (Crawford, 1971; Bingham, 1971). The temperature data utilized in this study was obtained from the official

U. S. Department of Commerce Local Climatological Data Reports for the winter months of December, January, February 1960-1969 for Austin, Texas. The data was recorded at three hour intervals measured at the Municipal Airport, Austin, Texas.

II. The Problem Statement and Modeling Analysis

2.1 The Problem Objective

The specific objective of this study is to develop a model to forecast local station dry bulb temperature through the technique of adaptive estimation. The statistical evaluation of the modeling is to be the minimization of the mean-square-error (MSE) estimates of the system states (Jazwinski, 1970). The overall performance evaluation is to be a ratio comparison between the MSE of the system states and the local station persistence variance, where persistence is defined as a forecast that the future temperature state will be the same as the present state.

2.2 The Filter Algorithm

The Adaptive Kalman filter is utilized for model identification, classification, and process prediction. The model identification and classification utilizes a training set of data of approximately 360 samples, and process prediction, a test set of the same length.

State variable model representation is used throughout in that state variable modeling encompasses the concept of the state of a system. The state of a system separates the future from the past such that the state contains all the relevant information concerning the past history of the system required to determine the response for any input (DeRusso, 1967).

The model in the Adaptive filter (Figure 1), which specifies each Kalman filter (matched to each admissible value of θ) for filtering the state of a Gauss-Markov process with unknown parameters, is characterized by two difference equations (Sims, 1969):

$$x(k) = \Phi(k/k-1/\theta_i)\, x(k-1) + D(k-1/\theta_i)\, u(k-1) \qquad (2.1)$$

$$z(k) = H(k/\theta_i)\, x(k) + v(k) \qquad (2.2)$$

where $x(\cdot)$ is the n x 1 state vector; $z(\cdot)$ is the r x 1 output data vector; $\Phi(\cdot)$ is the n x n state transition matrix, conditioned on θ_i; $D(\cdot)$ is the n x m input matrix, conditioned on θ_i; $H(\cdot)$ is the r x n measurement matrix, conditioned on θ_i; and $u(\cdot)$ and $v(\cdot)$ are the n x 1 input excitation and r x 1 measurement error vectors, respectively, of Gaussian zero mean white noise processes.

The modeling of the random process in Eqs. (2.1) and (2.2) assumes that $E[x(0)\, v^T(k)] = 0$, $E[x(0)\, u^T(k)] = 0$, for all k and that x(0) is Gaussian.

The parameter value, θ_i, is a point of a finite dimensional vector space having a dimension equal to the number of unknown model parameters, which will prevent the precise characterization of the Kalman filtering process (Hilborn, 1968, and Sims, 1969).

The notation λ_k represents all observations through time k. The "weighting" probabilities $p(\theta_i/\lambda_k)$ are given recursively by:

$$P(\theta_i/\lambda_k) = \frac{|P(k/k-1;\theta_i)|^{-\frac{1}{2}} \exp\left[-\tfrac{1}{2}\tilde{z}^T(k/k-1;\theta_i)P_z^{-1}(k/k-1;\theta_i)\tilde{z}(k/k-1;\theta_i)\right]P(\theta_i/\lambda_{k-1})}{\sum_{j=1}^{L} |P(k/k-1;\theta_j)|^{-\frac{1}{2}} \exp\left[-\tfrac{1}{2}\tilde{z}^T(k/k-1;\theta_j)P_z^{-1}(k/k-1;\theta_j)\tilde{z}(k/k-1;\theta_j)\right]P(\theta_j/\lambda_{k-1})} \tag{2.3}$$

where

$$P_z(k/k-1;\theta_i) = H(k/\theta_i)P(k/k-1;\theta_i)\ H^T(k/\theta_i) + R(k/\theta_i), \tag{2.4}$$

$$\tilde{z}(k/k-1;\theta_i) = z(k) - \hat{z}(k/k-1;\theta_i), \text{ and} \tag{2.5}$$

$$\hat{z}(k/k-1;\theta_i) = H(k/\theta_i)\ \Phi(k,k-1/\theta_i)\ \hat{x}(k-1/k-1;\theta_j). \tag{2.6}$$

The terms $R(k/\theta_i)$ and $P_z(k/k-1;\theta_i)$ are the measurement noise covariance and error covariance of z respectively (Sims, 1969). The terms $P(\theta_i/\lambda_0) = P(\theta_i)$ for i = 1, 2, ..., L, are known constants which represent the a-priori confidence in the θ_i parameter values (Sims, 1969).

The error covariance of z is obtained recursively from the discrete version of the Riccati equation (Meditch, 1969).

$$\dot{P} = F(t)P + PF^{T}(t) - PH^{T}(t)R^{-1}(t)H(t)P + K(t)Q(t)K^{T}(t) \tag{2.7}$$

and is given by (Sims, 1969)

$$P(k/k-1;\theta_i) = \Phi(k,k-1/\theta_i)\,P(k-1/k-1;\theta_i)\,\Phi^{T}(k,k-1/\theta_i) + D(k-1/\theta_i)\,D^{T}(k-1/\theta_i) \tag{2.8}$$

$$P_z(k/k-1;\theta_i) = H(k/\theta_i)\,P(k/k-1;\theta_i)H^{T}(k/\theta_i) + R(k/\theta_i) \tag{2.9}$$

$$P(k/k;\theta_i) = P(k/k-1;\theta_i) - K(k,k/\theta_i)P_z(k/k-1;\theta_i)K^{T}(k,k/\theta_i) \tag{2.10}$$

$$K(k,k/\theta_i) = P(k/k-1;\theta_i)H^{T}(k/\theta_i)\,P_z^{-1}(k/k-1;\theta_i) \tag{2.11}$$

$$\hat{x}(k/k;\theta_i) = \Phi(k,k-1/\theta_i)\hat{x}(k-1/k-1;\theta_i) + K(k,k/\theta_i)\tilde{z}(k/k-1;\theta_i) \tag{2.12}$$

The adaptive prediction of the system state is formed by summing elemental conditional state predictions which are weighted by the conditional probabilities of the parameter values given the data. That is,

$$\hat{x}(k/k-1) = \sum_{j=1}^{L} \hat{x}(k/k-1;\theta_j)\,p(\theta_j/\lambda_k) \tag{2.13}$$

where

$$\hat{x}(k/k-1;\theta_j) = \Phi(k/k-1;\theta_j)\,\hat{x}(k-1/k-1;\theta_j) \tag{2.14}$$

Three initial conditions are required: $p(\theta_i / \lambda_0)$, $x(0/0)$, and $P(0/0)$ to perform the structural algorithm. The probability, $p(\theta_i / \lambda_0)$ is chosen to be $1/L$ so as not to establish an initial model bias for θ_i. The initial filter conditions $x(0/0)$ and $P(0/0)$ may be chosen arbitrarily to be the zero vector and identity matrix of proper dimension, respectively.

The Adaptive filter provides optimal mean-square-error estimates of the system states when the state transition matrix, the measurement matrix, the measurement noise covariance matrix, and the initial conditions are not completely known. That is, the above listed parameters are known to within a finite set of known parameter vectors.

The optimal estimate of the system states is obtained by minimization of the error covariance matrix, $P(k/k) = E[\tilde{x}(k/k)\,\tilde{x}^T(k/k)]$ in Eq. (2.10). The Adaptive filter is an improvement upon the ordinary Kalman-Bucy filter in that it allows incomplete specification of the system model. When a Kalman filter processes data which is generated from a system model that is adequately characterized by a set of dynamical and statistical parameters different from those used in the filter, suboptimal estimation of the system state occurs. In this case, the degraded performance of the Kalman filter must be accepted or techniques must be employed which permit the filter to adapt to the unknown parameters. The Adaptive Kalman filter is ideally suited for this purpose.

Thus, Magill (1965) and Lainiotis (1971a, b, 1968) have provided a theoretical tool which satisfies the salient statistical requirements of the temperature process; i.e., the Adaptive Kalman filter provides optimal estimates of the system states for a nonlinear, nonstationary, Gauss-Markov process with unknown model parameter values. Additionally, the Adaptive filter, through state variable vector modeling, allows non-scalar state implementation; hence, the order of the system need not be specified a-priori. This latter point is extremely important if the temperature process is Markov-2 or greater.

2.3 Model Identification and Classification

The model identification and classification method on the training set employes the technique outlined by Sengbush (1969) which is to:

a. Choose, arbitrarily, a coarse model (single Kalman filter).

b. Search with one parameter element while holding all the model values constant.

c. Obtain a local minimum (utilizing the Adaptive filter) with the first element, fix this value and choose a second, repeating the method until all model elements are exhausted. This completes one repetition of the model search.

d. Store the parameter values in memory and use this for the next search.

The classification of the system model (classification dimension of the state vector $x(\cdot)$) utilizes the prediction results on the training set for models obtained by the technique outlined above for specific dimensions of the state vector $x(\cdot)$. That is, start with $n = 1$ (scalar state vector), search for the model element values, predict the training set utilizing the Adaptive filter (Eq. 2.13), and observe the MSE, increase n to $n + 1$, and repeat the process. The order of the system model is obtained when the prediction MSE reaches a minimum level value.

The application of the above procedure for the temperature process utilizing three hour sampled data was:

Dimension of $x(\cdot)$, $n =$	Training Set MSE	Persistence MSE	Variance Ratio
1	65.6	44.7	1.470
2	16.8	44.7	0.375
3	17.0	44.7	0.380
4	16.9	44.7	0.378

The variance ratio is the ratio of the training set MSE to the persistence MSE. This result indicates that the temperature process is adequately modeled by a state vector of dimensionality $n = 2$.

In addition to the comparison of prediction results of the Adaptive filter to the persistence forecast, two additional filtering schemes were applied. These were standard Least-Square and straight line projection predictions (Robinson, 1967) on the same training set. The results were:

Method	Training Set MSE	Variance Ratio
Least-Square	26.4	0.590
Straight-Line	26.0	0.582

It is interesting to note that the above two methods significantly reduced the error below the persistence forecast but did not perform as well as the Adaptive filter.

Verification of the model parameter identification was done by checking the statistics of the "innovation process", $\tilde{z}(\cdot)$; i.e., the error process of the prediction: $\tilde{z}(\cdot) = z(\cdot) - \hat{z}(\cdot)$. The error process, $\tilde{z}(\cdot)$, for a correctly fitted model is a zero mean, white noise process (Astrom, 1965; Frost, 1971). Checking the "innovation" of the n = 2 Adaptive filter model indicates that the mean value of $\tilde{z}(\cdot)$ was approximately zero (equals - 0.02) and had an autocorrelation shown in Figure 2, which very nearly approximates an impulse function. Figure 3 is autocorrelation of a computer simulated white noise process which is used for comparison purposes.

The identified model (n = 2) for the Adaptive filter is (maximum aposteriori probability criterion):

$$\Phi = \begin{bmatrix} 0.3205 & 0.5357 \\ 0.5357 & 0.5521 \end{bmatrix} \qquad H = [1\ 0]$$

$$D = \begin{bmatrix} 4.0737 \\ 13.7199 \end{bmatrix} \qquad R = [0.9089]$$

Transferring the above result from a three hour (one step) to a γ step prediction (for example: twenty-four hour prediction, $\gamma = 8$), is easily accomplished by the relationship (De Russo, 1967):

$$x(k + \gamma/k) = \Phi(k + \gamma/k)\, x(k/k) \tag{2.15}$$

where

$$\Phi(k + \gamma/k) \cong [\Phi(k + 1)/k]^{\gamma} \tag{2.16}$$

III. Test Set Prediction

Quantizing about the element values for $\Phi(\cdot)$, $D(\cdot)$, $H(\cdot)$, and $R(\cdot)$ and using Eq. (2.15) for the n = 2 model with $\gamma = 8$, a MSE result of 118.2 on the test set was obtained with a corresponding persistence MSE of 162.2. This result was obtained utilizing thirty filters in the bank of Kalman filters of Figure 1. This twenty-four hour prediction gives a variance ratio of 0.725.

IV. Conclusions and Recommended Extensions

The prediction results on the test set, while not competitive for immediate operational use, do establish the feasibility and efficacy of the adaptive nonlinear estimation approach with the Adaptive Kalman filter. It is anticipated that the incorporation of correlating information from other geographic stations, atmospheric, and other surface data (dew point, relative humidity, etc.) will significantly improve the prediction performance of the Adaptive filter. These extensions are to be incorporated as additional elements in r x 1 output data vector, z (·); and, hence, require no new theoretical development in the filtering theory. Computer programs are found in Penrose (1973).

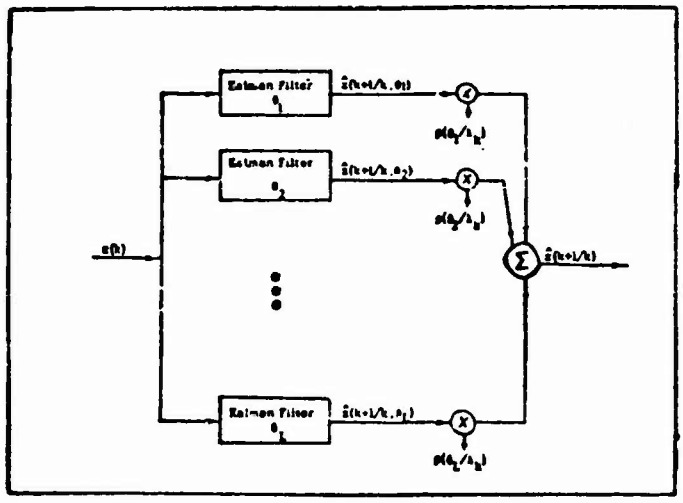


Figure 1 - Adaptive Kalman Filter

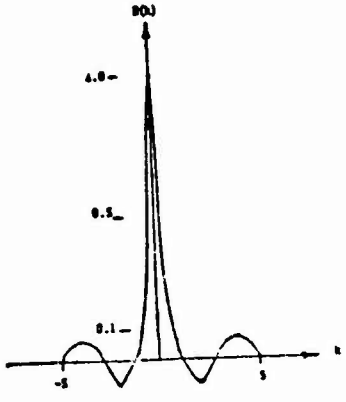


Figure 2 - Autocorrelation of $\tilde{z}$ Process

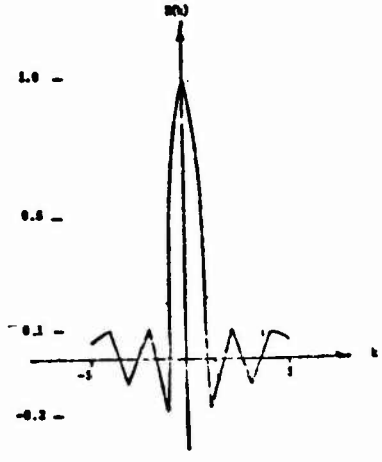


Figure 3 - Autocorrelation of White Noise Process

REFERENCES

Astrom, K., and Bohlin, T., 1965, "Numerical Identification of Linear Dynamic Systems from Normal Operating Records", Proceedings Second IFAC Symposium on Theory of Self-Adaptive Systems, pp. 96-111.

Bingham, C., 1971, "Distributions of Weekly Averages of Diurnal Temperature Means and Ranges about Harmonic Curves", Monthly Weather Review, Washington, D. C. 89(9): 357-367.

Bucy, R. S., 1965, "Nonlinear Filtering Theory", IEEE Trans. on Automatic Control, Vol. AC-10, p. 198.

Crawford, K. C., 1971, "Customer Tailored Forecasts Using Markov Chains and Decision Theory", International Symposium on Probability and Statistics in the Atmospheric Sciences, Honolulu, June 1-4, 1971, Preprints of Papers, Boston - American Meteorological Society, pp. 100-105.

De Russo, P. M., et al, 1967, State Variables for Engineers, New York, New York, John Wiley and Sons, Inc.

Frost, P. A., and Kailath, T., June 1971, "An Innovations Approach to Least-Squares Estimation - Part III: Nonlinear Estimation in White Gaussian Noise", IEEE Trans. on Automatic Control, Vol. AC-16, No. 3, pp. 217-226.

Hilborn, C. G., and Lainiotis, D. G., May 1968, "Recursive Computations for the Optimal Trading of Time-Varying Parameters", IEEE Trans. on Information Theory, Vol. IT-14, No. 3, pp. 514-515.

Jazwinski, A. H., 1970, Stochastic Processes and Filtering Theory, New York, New York, Academic Press.

Jones, R. H., 1971, "Spectrum Estimation and Time Series Analysis-A Review", International Symposium on Probability and Statistics in Atmospheric Sciences.

Lainiotis, D. G., April 1968, "A Nonlinear Adaptive Estimation Recursive Algorithm", IEEE Trans. on Automatic Control, Vol. AC-13, pp. 197-198.

Lainiotis, D. G., 1971a, "Optimal Nonlinear Estimation", International Journal of Control, Vol. 14, No. 6, pp. 1137-1148.

Lainiotis, D. G., 1971b, "Optimal Adaptive Estimation: Structure and Parameter Adaptation", IEEE Trans. on Automatic Control, Vol. AC-16, pp. 160-170.

Magill, D. T., "Optimal Adaptive Estimation of Sampled Stochastic Processes", IEEE Trans. on Automatic Control, October 1965, Vol. AC-10, No. 4, pp. 434-439.

Meditch, J. S., 1969, Stochastic Optimal Linear Estimation and Control, McGraw-Hill Company.

Penrose, N. B., 1973, Application of Adaptive Estimation to Weather Forecasting, Ph.D. Dissertation, University of Texas at Austin, Austin, Texas.

Robinson, E. A., 1967, Multichannel Time Series Analysis with Digital Computer Programs, San Francisco, California, Holden-Day.

Sengbush, R. L., and Lainiotis, D. G., August 1969, "Simplified Parameter Quantization Procedure for Adaptive Estimation", IEEE Trans. on Automatic Control, Vol. AD-14, No. 4, pp. 424-425.

Sims, F. L., and Lainiotis, D. G., April 1969, "Recursive Algorithms for The Calculation of the Adaptive Kalman Filter Weighting Coefficients", IEEE Trans. on Automatic Control, Vol. AC-14, No. 2, pp. 215-218.

ROBERT H. HAVESON
AMMUNITION DEVELOPMENT AND ENGINEERING DIRECTORATE
ARMAMENT COMMAND
PICATINNY ARSENAL
DOVER, NEW JERSEY

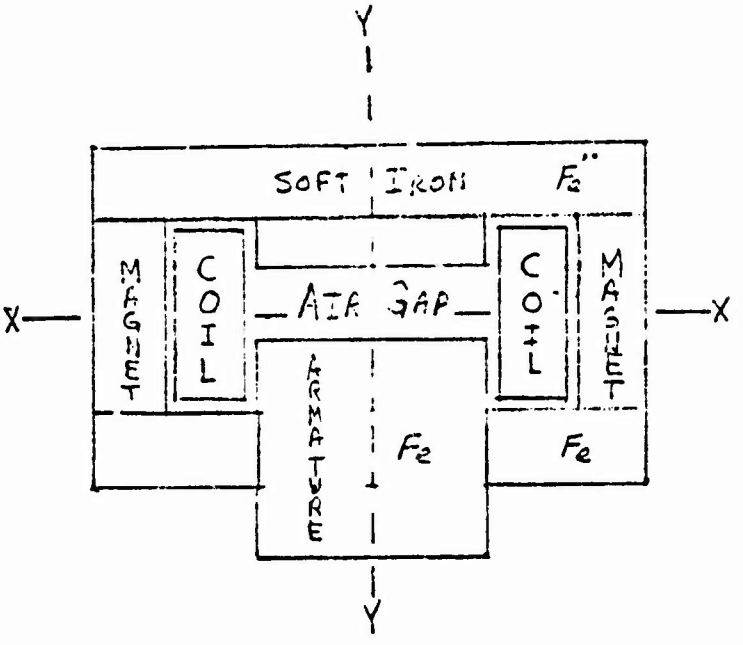
INTRODUCTION

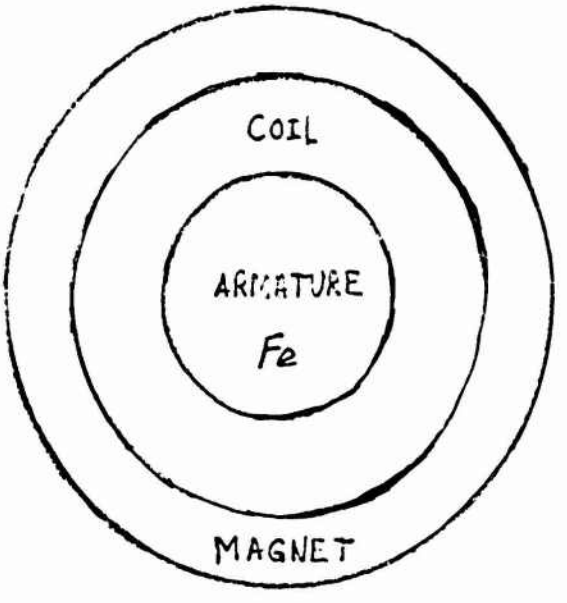
Magnetic circuits have been analyzed using methods which depend heavily on engineering judgment and empirical data. These methods have used elementary formulae and basic principles and have resulted in relatively good approximations. However, due to the solutions'great dependence on the engineer's practical guesses, these methods are limited when used to predict effects caused by design changes.

The problem that this technique was applied to is the modelling of an electromagnetic generator (Figure I) used in the Advanced Beehive Electronic Fuze. The generator produces a voltage output when exposed to high acceleration forces. The generator contains a magnetic circuit which consists of a cylindrical magnet, soft iron top and bottom plates with a movable soft iron armature through the center. The armature has a coil of wire surrounding it. When acceleration is experienced, the armature is moved, an air gap developed, and due to a change in flux, a voltage is induced into the coil. The goal of this analysis is to determine magnitudes and patterns of the developed flux. The approach chosen was to develop the model of the configuration where the generator had no gap. If the model would handle this situation, it is safe to assume that the technique handles different gaps as well as reasonable changes in the geometry.

This report describes a computer method which solves the partial differential equation which is derived from Maxwell's equation for an electromagnetic generator.

FIGURE I





MAGNETIC GENERATOR

$$\frac{\partial}{\partial Z}\left(V \frac{\partial A}{\partial Z}\right) + \frac{\partial}{\partial r}\left(\frac{V}{r} \frac{\partial (rA)}{\partial r}\right) = 0 \qquad (1)$$

The differential equation is stated in its finite difference form. (Figure II).

Where V is the reluctivity of the material, A is the magnetic vector potential and h is the distance between the points.

THE PROCEDURE

Since the equation is in cylindrical coordinates and there is symmetry about the center vertical line, the system is dependent on r and Z only. The system can be described by a two coordinate grid of I ordinate and J abscissa (Figure III).

Instead of the right-hand side of the difference equation being set to zero, it is set equal to R_o where R_o is the residue in excess of zero. For the first iteration as good a guess as possible is made for the V and A. After each residue is calculated during one iteration a new value for each magnetic vector potential (A(I,J)) is determined from the formula for relaxation.

$$A(I,J)_{K+1} = A(I,J)_K + \alpha \frac{R_o}{XD} \qquad (3)$$

where K is the iteration number, α is the factor of overrelaxation and XD is a function of the reluctivities surrounding A(I,J).

The magnetic flux density is determined from the relationships

$$B_r = \left(-\frac{\partial A}{\partial Z}\right) \qquad (4)$$

$$B_Z = \frac{1}{r} \frac{\partial (rA)}{\partial r} \qquad (5)$$

NUMERICAL APPROACH

FINITE DIFFERENCE EQUATION

$$R_o=\frac{1}{h_2+h_4}\left[\left(\frac{\nu_{II}+\nu_{III}}{2}\right)\left(\frac{A_2-A_o}{h_2}\right)-\left(\frac{\nu_I+\nu_{IV}}{2}\right)\left(\frac{A_o-A_4}{h_4}\right)\right]+$$

$$\frac{1}{h_1+h_3}\left[\frac{(\nu_I+\nu_{II})}{2(r+\frac{h_1}{2})}\left[\frac{(r+h_1)A_1-rA_o}{h_1}\right]-\right. \tag{2}$$

$$\left.\left(\frac{\nu_{III}+\nu_{IV})}{2(r+\frac{h_3}{2})}\left[rA_o-(r-h_3)A_3\right]\right]\right]$$

EXTRAPOLATED LIEBMANN METHOD

$$ANEW=AOLD+\alpha\frac{R_o}{XD} \qquad XD=f(\nu, h)$$

FIGURE II

The difference form

$$B_r = -\left[\frac{A_o + A_1 - A_4 - A_8}{2h_3}\right]\bar{a}_r \tag{6}$$

$$B_Z = \frac{(r+h)\left(A_1 + A_8\right) - r\left(A_o + A_4\right)}{2(r+h_1/2)h_1}\bar{a}_Z \tag{7}$$

$$B = \sqrt{B_r^2 + B_Z^2} \tag{8}$$

The reluctivities, V, are calculated from

$$V = \frac{H}{\bar{B}} \tag{9}$$

where H is the field intensity of the region being considered. In this problem the magnet material is Alnico V and H was determined from an equation. The newly calculated potentials and reluctivities are substituted into the difference equation and the cycle is repeated.

This procedure is continued until the residues fall below a prespecified epsilon, in which case the solution is obtained. To determine if the grid size is optimum, the number of points are increased and the iterative process is repeated until there is little difference between adjacent solutions.

FLUX PATTERNS

It can be shown that the curves EQ = rA = constant are the lines of flux. If these equipotentials are plotted, the resulting curves describe the flux pattern of the magnetic system. The points of EQ are determined simply from the

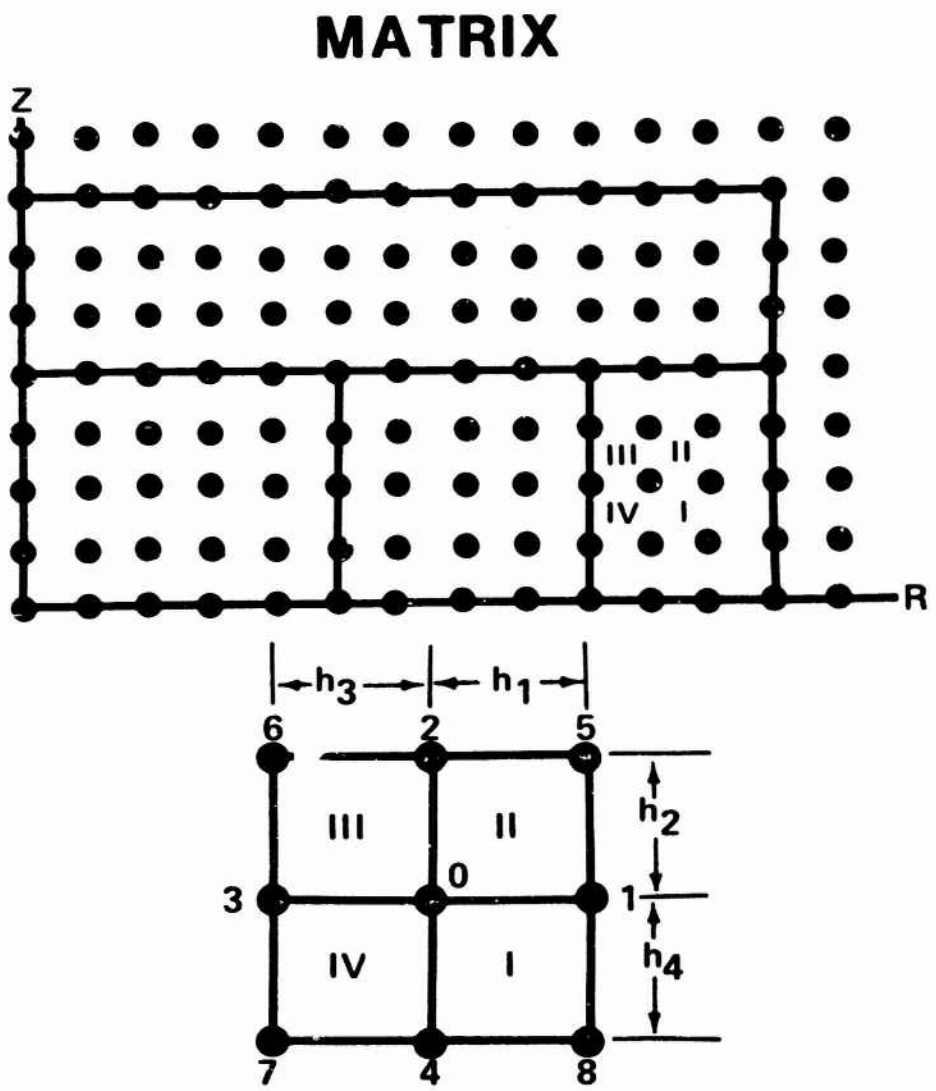


FIGURE III

product of each grid point (A(I,J) and its radial distance r from the vertical axis of symmetry (Figure IV).

ACCURACY

Accuracy is limited by the closeness of the difference equation to the actual partial differential equation. It is also affected by the algorithm and most significantly by the number of points which define the system. When it is necessary to increase the number of points to improve the accuracy, the most efficient method would be to do so only where necessary. Generally this would be along boundaries. One useful method to determine where problem areas lie would be to take advantage of the problem's symmetry. Consider changing the problem as a test in order to create new symmetry and again eliminate points which must be iterated.

In this problem, the case where there was no air gap was considered, thereby including top and bottom symmetry as well as side by side symmetry (Figure III).

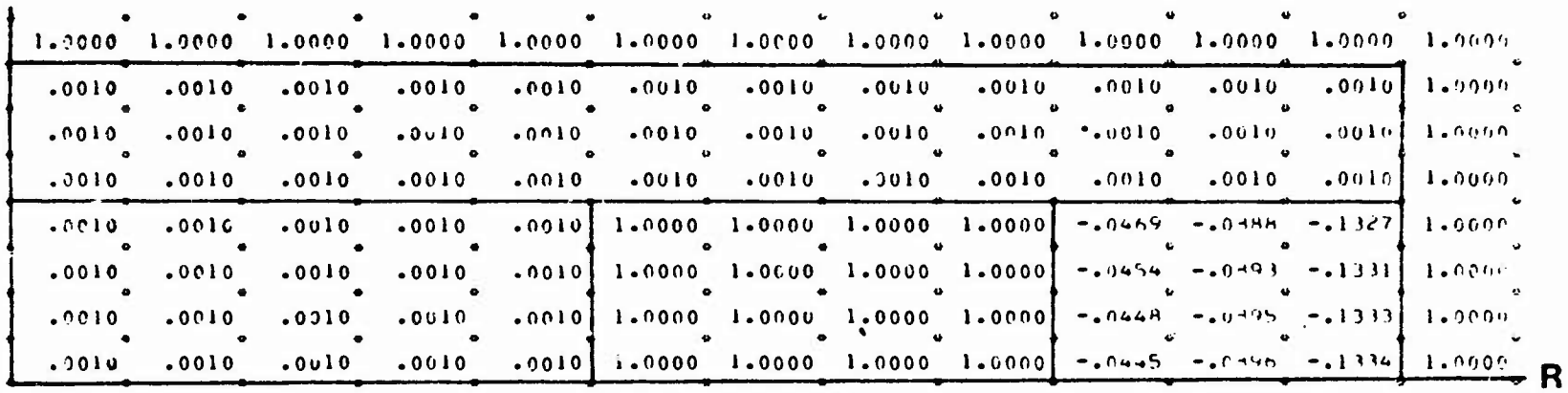
BOUNDARY CONDITIONS

The difference equation will satisfy the boundary conditions everywhere such that H tan and B norm are continuous at the boundaries (Figure V). Each cell is a region of reluctivity bounded on four sides by other reluctivities. At boundaries where large changes in the path of the flux lines occur, it is necessary to employ a finer grid. A finer grid consists of increasing the number of points uniformly in the required regions. This procedure adds complexity

MAGNETIC GENERATOR SOLUTION

NON-UNIFORM RELUCTIVITIES (ν)

RELUCTIVITY



MAGNETIC VECTOR EQUIPOTENTIALS (EQ)

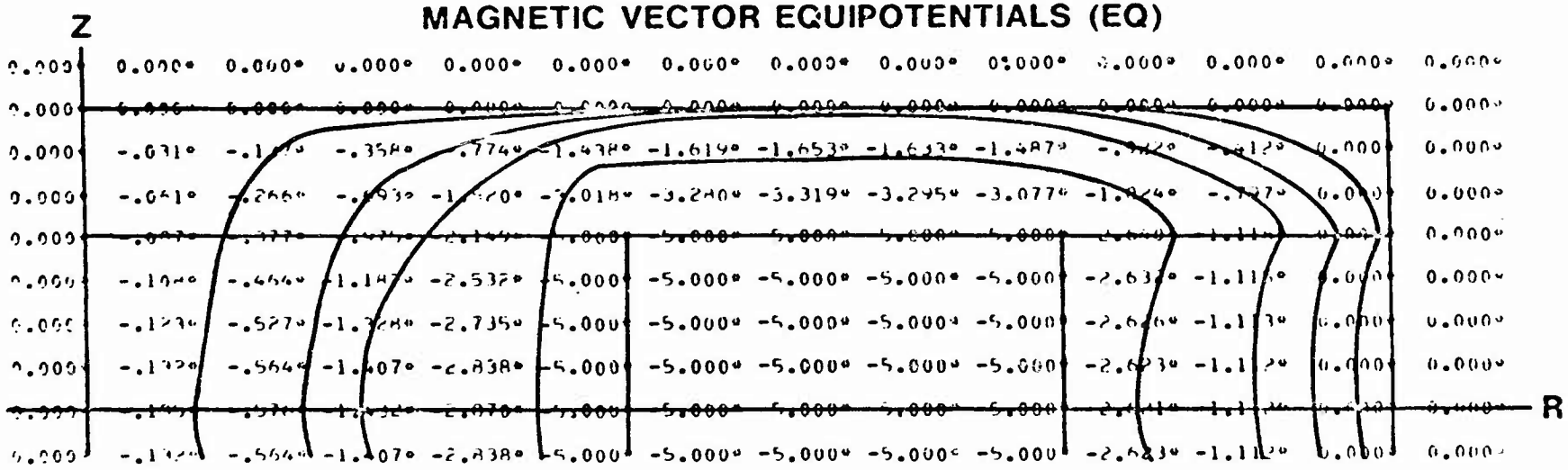
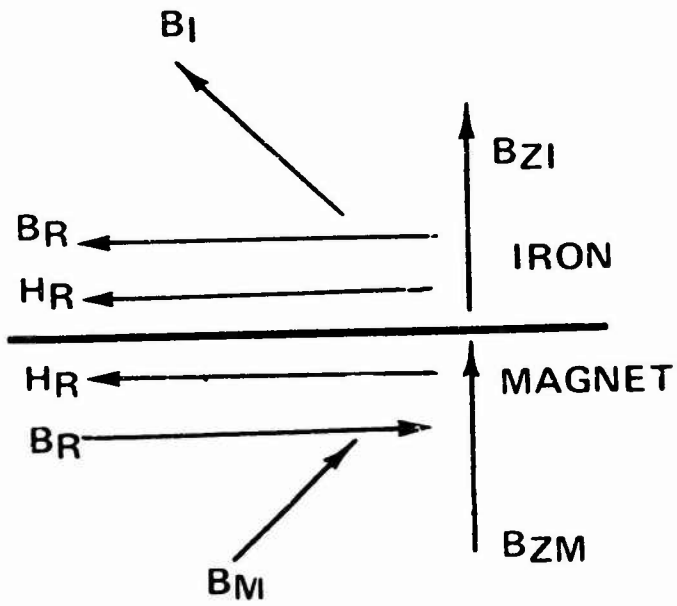
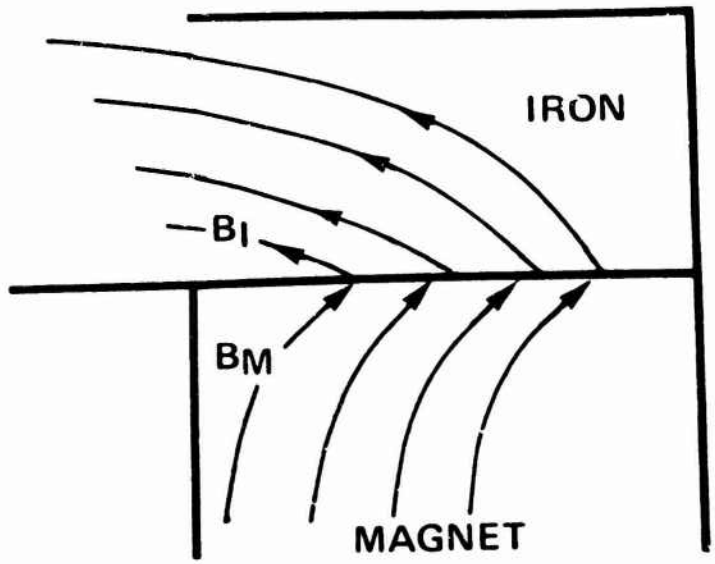


FIGURE IV

BOUNDARY FLUX LINES





AT BOUNDARY:

$B_{ZI}=B_{ZM}$

IF B_{RI} MOVES TO LEFT H_{RI} MOVES TO RIGHT

$H_{RM}=H_{RI}$

THEN B_{RM} MOVES TO RIGHT SINCE $B_{RM}=H_M/V_M$ AND H_M AND V_M ARE NEG

FIGURE V

to the algorithm, however. An appreciable computer cost saving will be achieved when selectively increasing only the number of points which are located on the boundaries.

EQUATION TEST

In order to develop confidence in the correctness of a finite difference solution it is imperative that a test should be devised. This test should result in a one to one comparison between an analytic solution and the iterative method. Any test should include as many as possible of the complexities which are to be found in the actual problem. Since more often than not, real world problems cannot be completely tested by this method a certain amount of trial and error procedures must be resorted to.

TEST #1

The first comparison test was with an exact solution derived from equation (1) where

$$A(r,Z) = \frac{I\left(\frac{\pi r}{L}\right)}{I_1\left(\frac{\pi a}{L}\right)} \sin \frac{\pi Z}{L} \qquad (10)$$

where I_1 refers to a bessel function.

The iterative method compared favorably with the exact solution.

THE COMPUTER PROGRAM

The flow chart of Figure VII describes the FORTRAN IV program. The most significant steps are:

(1) Read geometry dimensions.

(2) Read data which is necessary in order to determine approximate magnetic vector potential values for the initial iteration.

(3) Determine the initial matrix for the reluctivities V.

(4) Begin the iteration process and calculate new potential values A.

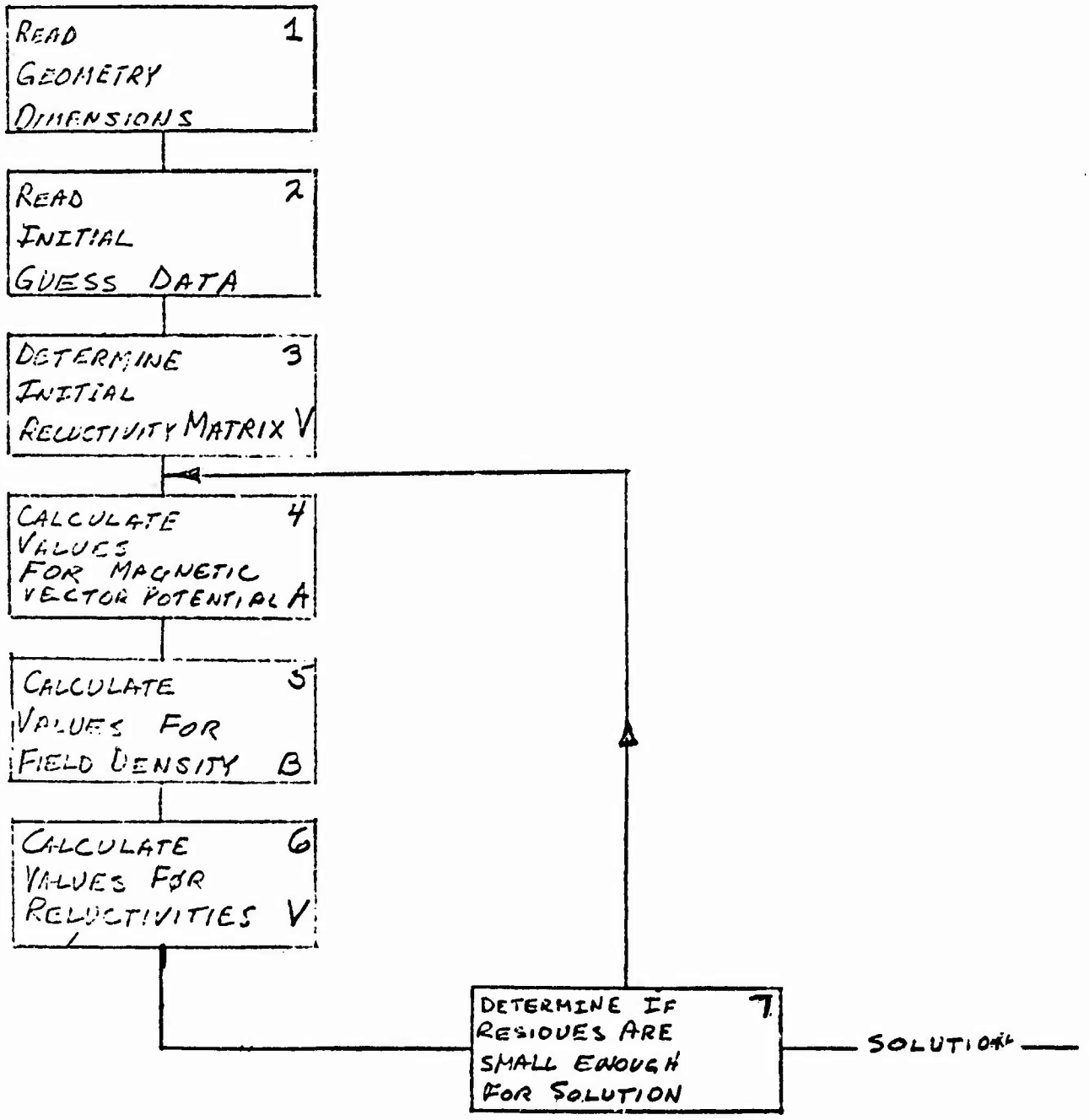
(5) Calculate values for the flux density B.

(6) Calculate values for the reluctivities V.

(7) Return to step 4 and repeat process until residues have been reduced to acceptable values.

FIGURE VII

THE COMPUTER PROGRAM:



CONCLUSION

The program has obviously shown its capability to solve problems such as that of Test 1 and the magnetic generator problem. However, in the generator problem the residues converged close to the solution but for no immediately obvious reason proceeded to diverge. At the critical point where this divergence occurred, the values of the parameters were meaningful when considering engineering judgment. Closer examination has shown that while the boundary conditions have been satisfied by equation (1) the flux lines upon crossing the soft iron-magnet boundary have a sharp change in direction. In order for the program to correctly resolve this condition, the number of points on this boundary must be increased.

This technique described above shows promise for solutions to physical problems which at times are considered impossible or impractical to attempt.

REFERENCES

(1) P. Lorrain and Dale Corson, "Electromagnetic Fields and Waves", San Francisco: Freeman, pp. 400-414, 1970.

(2) C. A. Holt, "Introduction to Electro-magnetic Fields and Waves", New York: Wiley, 1967.

(3) K. J. Binns and P. J. Lawrenson, Analysis and Computation of Electric and Magnetic Field Problems", New York: Pergamon Press, 1963.

(4) P. Silvester, "Modern Electromagnetic Fields", Englewood Cliffs: Prentice Hall, 1968.

(5) S. V. Ahamed and E. A. Erdelyi, "Nonlinear Theory of Salient Pole Machines", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-85, pp. 61-70, January 1966.

PERTURBED KUHN-TUCKER POINTS AND RATES OF CONVERGENCE FOR A CLASS OF NONLINEAR-PROGRAMMING ALGORITHMS

Stephen M. Robinson
Mathematics Research Center
University of Wisconsin-Madison

ABSTRACT. This paper establishes quantitative bounds for the variation of an isolated local minimizer for a general nonlinear program under perturbations in the objective function and constraints. These bounds are then applied to establish rates of convergence for a class of recursive nonlinear-programming algorithms.

1. INTRODUCTION. In recent years a considerable amount of effort has been expended in analysis of the behavior of the optimal set of a mathematical program under perturbations in the objective function and/or the constraints. A general treatment of this question is given in [1, pp. 115-117], and other works in the same area include [2, 5, 15]. These analyses all investigate the question of continuity; they do not give any quantitative bounds for changes in the solution set. For a general nonlinear program, obtaining such quantitative bounds for the entire solution set appears to be a very difficult problem. In Section 2 of this paper we take a different point of view: instead of considering the entire solution set, we investigate the behavior of an isolated local minimizer of a not-necessarily-convex nonlinear program when the objective function and the constraints are perturbed. An approach due to Fiacco and McCormick [7] shows that under reasonable conditions the study of such a minimizer reduces to the study of the locally unique zero of a certain system of nonlinear equations. An immediate consequence of this fact is that one may apply the implicit-function theorem to obtain, not only a proof of the existence and local uniqueness of the minimizer for slightly perturbed problems, but also quantitative bounds on its variation, and on the variation of the associated Lagrange multipliers, in terms of quantities associated with the problem functions (objective and constraints). These quantitative bounds are then applied in Section 3 to derive convergence rates for a class of nonlinear-programming algorithms which compute successive approximations to Kuhn-Tucker points by replacing the original problem by a sequence of approximate problems, each having a simpler structure than the original problem.

2. PERTURBATION OF KUHN-TUCKER POINTS. In [7, §2.4, Th. 6], Fiacco and McCormick gave an excellent discussion of the behavior of Kuhn-Tucker points when the data of the problem contain linear perturbations. We extend their results here to show that quite general perturbations can be treated by the same technique; in addition, we obtain the quantitative bounds mentioned in Section 1.

Sponsored by the United States Army under Contract No. DA-31-124-ARO-D-462.

We assume knowledge of the first-order Kuhn-Tucker conditions of nonlinear programming; we shall require also a knowledge of the second-order sufficiency conditions [7, §2.3]. For the program

$$\underset{x}{\text{minimize}} \ \{\theta(x) \mid g(x) \leqq 0, \ h(x) = 0\} \tag{1}$$

these are said to hold at a point $(\bar{x}, \bar{u}, \bar{v})$, where $\bar{u}$ and $\bar{v}$ are vectors of multipliers for g and h respectively, if $(\bar{x}, \bar{u}, \bar{v})$ is a first-order Kuhn-Tucker point of (1) and if in addition, for each $x \neq \bar{x}$ such that

$$g_i'(\bar{x})(x - \bar{x}) = 0 \quad \text{for all } i \text{ with } \bar{u}_i > 0,$$

$$g_i'(\bar{x})(x - \bar{x}) \leqq 0 \quad \text{for all } i \text{ with } g_i(\bar{x}) = 0 \text{ and } \bar{u}_i = 0,$$

and

$$h_j'(\bar{x})(x - \bar{x}) = 0 \quad \text{for all } j,$$

we have

$$\mathfrak{L}_{11}''(\bar{x}, \bar{u}, \bar{v})(x - \bar{x})^2 > 0,$$

where

$$\mathfrak{L}(x, u, v) := \theta(x) + u^T g(x) + v^T h(x)$$

Here and later in the paper, we use primes to indicate (Fréchet) derivatives, with subscripts on partial derivatives to indicate the arguments with respect to which the differentiation is to be performed.

The conditions just given are sufficient for $\bar{x}$ to be an isolated local minimizer for (1) [7, §2.3, Th. 4]. We shall also require the concept of strict complementary slackness, which is said to hold at $(\bar{x}, \bar{u}, \bar{v})$ if for each i either $\bar{u}_i > 0$ or $g_i(\bar{x}) < 0$.

The theorem stated below employs functions θ, g and h whose arguments are (x, p) rather than (x) as in (1). The quantity p is to be interpreted as a perturbation. It will be permitted to vary around a fixed value $\bar{p}$; the arguments $(x, \bar{p})$ can be thought of as those of the "unperturbed" problem. We introduce the perturbed nonlinear program

$$\underset{x}{\text{minimize}} \{\theta(x, p) \mid g(x, p) \leq 0, \ h(x, p) = 0\}, \qquad (1\{p\})$$

the perturbed Lagrangian

$$\mathcal{L}(x, u, v, p) := \theta(x, p) + u^T g(x, p) + v^T h(x, p),$$

and a function f defined by

$$f(x,u,v,p) := [\mathcal{L}_1'(x,u,v,p),\ u_1 g_1(x,p), \ldots, u_m g_m(x,p),\ h_1(x,p), \ldots, h_q(x,p)]^T$$

with which we shall be concerned in what follows. Note that f comprises the equalities of the Kuhn-Tucker conditions for $(1\{p\})$. The norm used on $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q$ is arbitrary, but is fixed throughout the paper.

THEOREM 1: Let P be a Banach space, let $\Gamma \subset \mathbb{R}^n$ and $\Pi \subset P$ be open sets, and let $\theta(x, p)$, $g(x, p)$, and $h(x, p)$ be functions from $\Gamma \times \Pi$ into $\mathbb{R}$, $\mathbb{R}^m$, and $\mathbb{R}^q$ respectively, all having second partial derivatives with respect to x which are jointly continuous on $\Gamma \times \Pi$. Let $\bar{p} \in \Pi$, and suppose that $\bar{x} \in \Gamma$ and some $\bar{u} \in \mathbb{R}^m$ and $\bar{v} \in \mathbb{R}^q$ form a Kuhn-Tucker triple of $(1\{\bar{p}\})$ at which the second-order sufficiency conditions are satisfied with strict complementary slackness and linear independence of the gradients to the active constraints.

Then there exist open neighborhoods $M(\bar{p}) \subset \Pi$ and $N(\bar{x}, \bar{u}, \bar{v}) \subset \Gamma \times \mathbb{R}^m \times \mathbb{R}^q$, and a continuous function $Z : M \to N$, such that $Z(\bar{p}) = (\bar{x}, \bar{u}, \bar{v})$ and for each $p \in M$, $Z(p)$ is both the unique Kuhn-Tucker triple of $(1\{p\})$ in N and the unique zero in N of the function $f(\cdot, \cdot, \cdot, p)$. Further, if $Z(p) =: (x(p), u(p), v(p))$ then for each $p \in M$, $x(p)$ is an isolated local minimizer of $(1\{p\})$ at which the second-order sufficiency conditions are satisfied with strict complementary slackness and linear independence of the gradients to the active constraints.

PROOF: We write z for $(x, u, v) \in \mathbb{R}^{n+m+q}$, $\bar{z}$ for $(\bar{x}, \bar{u}, \bar{v})$, $f(z, p)$ for $f(x, u, v, p)$, etc. The conditions placed upon $(\bar{x}, \bar{u}, \bar{v})$ and $\bar{p}$ are sufficient to ensure that the matrix $f_1'(\bar{z}, \bar{p})$ is nonsingular [10, p. 231]; here the subscript 1 denotes partial differentiation with respect to the argument z. Since $f(\bar{z}, \bar{p}) = 0$ because $\bar{z}$ is a Kuhn-Tucker triple of $(1\{\bar{p}\})$, the implicit-function theorem [8, Ths. 1-2 (4. XVII)] guarantees the existence of open neighborhoods $M_0(\bar{p}) \subset \Pi$ and $N_0(\bar{z}) \subset \Gamma \times \mathbb{R}^m \times \mathbb{R}^q$

and a continuous function $Z : M_0 \to N_0$, such that $Z(\bar{p}) = \bar{z}$ and for each $p \in M_0$, $Z(p)$ is the unique zero of $f(\cdot, p)$ in N_0. In addition, there are some open neighborhoods $N_1(\bar{z})$ and $M_1(\bar{p})$ such that for $(x, u, v) \in N_1$ and $p \in M_1$ we have for $1 \leq i \leq m$,

$$\left.\begin{array}{l} g_i(\bar{x}, \bar{p}) < 0 \text{ implies } g_i(x, p) < 0 \\ \text{and} \\ \bar{u}_i > 0 \quad \text{implies } u_i > 0 . \end{array}\right\} \qquad (2)$$

Let $N := N_0 \cap N_1$; then since Z is continuous and $Z(\bar{p}) = \bar{z}$, it is possible to find an open neighborhood $M_2(\bar{p}) \subset M_1 \cap M_0$ such that $p \in M_2$ implies $Z(p) \in N$. For any $p \in M_2$, the point $Z(p)$ satisfies the equalities of the Kuhn-Tucker conditions for $(\Pi\{p\})$ because $f(Z(p), p) = 0$. Let i be chosen with $1 \leq i \leq m$, and let $Z(p) = (\tilde{x}, \tilde{u}, \tilde{v})$. If $\bar{u}_i > 0$ then since $Z(p) \in N_1$ we have also $\tilde{u}_i > 0$, so $g_i(\tilde{x}, p) = 0$; on the other hand, if $g_i(\bar{x}, \bar{p}) < 0$ then $g_i(\tilde{x}, p) < 0$, so $\tilde{u}_i = 0$. Since strict complementary slackness holds at $(\bar{x}, \bar{u}, \bar{v})$ one of these two cases is applicable for each i. Hence $\tilde{u} \geq 0$ and $g(\tilde{x}, p) \leq 0$, so $Z(p)$ is a Kuhn-Tucker point of $(1\{p\})$; it is the only such point in N because it is the only zero of $f(\cdot, p)$ there.

To prove the final assertion of the theorem, we note that (2) implies that precisely the same inequality constraints will be active at $x(p)$, for each $p \in M_2$, as were active at $x(\bar{p})$, and that strict complementary slackness will hold for the triple $Z(p)$. Since the gradients to the constraints active at $x(\bar{p})$ were assumed to be linearly independent, and since by the continuity of Z, $g_i'(x(p), p)$ and $h_i'(x(p), p)$ will be continuous as functions of p for $p \in M_2$, it follows that there is an open neighborhood $M_3(\bar{p}) \subset M_2$ within which they remain a linearly independent set. But we observed just previously that these are also the gradients to the constraints active at $x(p)$ for <u>any</u> $p \in M_2$ (they will subsequently be called simply the "active gradients"), and this remains true for the smaller neighborhood M_3. Thus the only thing left to be shown is that the second-order sufficiency conditions remain valid. To prove this, we observe first that the first-order Kuhn-Tucker conditions are satisfied by each triple $Z(p)$. Also, if the set of active gradients contains n vectors, the second-order sufficiency conditions are trivially satisfied by $Z(p)$ for each $p \in M_3$, and there is nothing more to prove. We may therefore assume with no loss of generality that there are fewer than n active gradients. For each $p \in M_3$, let $G(p)$ be a matrix whose rows are those gradients evaluated at $(x(p), p)$, and consider the multivalued function

$$\Gamma(p) := \{w \in \mathbb{R}^n \mid G(p)w = 0\}.$$

It is easy to show that the graph of Γ is closed; thus, if we let S be the set of vectors in $\mathbb{R}^n$ having Euclidean length 1, the function $\Gamma(p) \cap S$ is upper semicontinuous on M_3. It is also nonempty there, because of our assumption about the number of active gradients. Now consider the functional

$$\delta(p) := \min_w \{w^T \mathcal{L}''_{11}(x(p), u(p), v(p), p)w \mid w \in \Gamma(p) \cap S\}.$$

Since the quantity being minimized is jointly continuous in w and p, and since the constraint set $\Gamma(p) \cap S$ is upper semicontinuous in p, it follows from [1, Th. 2, p. 116] that δ is lower semicontinuous in p for $p \in M_3$. But since the second-order sufficiency conditions hold for $Z(\bar{p})$, we have $\delta(\bar{p}) > 0$; hence for all p in some open neighborhood $M(\bar{p}) \subset M_3$, we also have $\delta(p) > 0$, and so for every $p \in M$ the triple $Z(p)$ satisfies the second-order sufficiency conditions. This completes the proof of Theorem 1.

In an independent but essentially simultaneous paper, Fiacco [6] has proved a result similar to Theorem 1. He makes stronger assumptions than we do (viz., that θ, g and h are each twice continuously differentiable in (x, p)), and consequently is able to prove stronger results. In particular, he proves that the function which we call $Z(p)$ is (under his assumptions) actually continuously differentiable, and he uses this fact to prove several useful and interesting results about sensitivity analysis. We prefer to avoid these stronger assumptions, but even so it is possible to find a quantitative bound on the variation of $Z(p)$. This is so because $Z(p)$ is a trajectory of zeros of the function $f(\cdot, p)$, and one can therefore use a well-known contraction technique to estimate its variation. We state this result as Theorem 2.

THEOREM 2: Assume the hypotheses and notation of Theorem 1. Then for any ϵ with $0 < \epsilon < 1$ there exist open neighborhoods $M_\epsilon(\bar{p}) \subset M$ and $N_\epsilon(\bar{x}, \bar{u}, \bar{v}) \subset N$ such that for any $\tilde{p} \in M_\epsilon$ and any $\tilde{z} \in N_\epsilon$ we have

$$\|\tilde{z} - Z(\tilde{p})\| \leq (1 - \epsilon)^{-1} \|f'_1(\bar{z}, \bar{p})^{-1}\| \, \|f(\tilde{z}, \tilde{p})\| .$$

PROOF: Let ϵ be chosen with $0 < \epsilon < 1$. Denote $\|f'_1(\bar{z}, \bar{p})^{-1}\|$ by β. The hypotheses ensure that the function $f'_1(z, p)$ is jointly continuous on $N \times M$; hence we can find an open neighborhood $M_4(\bar{p}) \subset M$ and a sufficiently small positive ν such that if $B(\bar{z}, \nu)$ denotes the open ball of radius ν about $\bar{z}$, we have $B(\bar{z}, \nu) \subset N$ and for each $p \in M_4$ and each $z \in B(\bar{z}, \nu)$,

$$\|f'_1(z, p) - f'_1(\bar{z}, \bar{p})\| \leq \epsilon \beta^{-1} .$$

Also, there is some λ with $0 < \lambda < \frac{1}{2}\nu$ and some open neighborhood $M_\epsilon(\bar{p}) \subset M_4$ such that for each $z \in B(\bar{z}, \lambda) =: N_\epsilon$ and each $p \in M_\epsilon$,

$$\|f(z, p)\| \leq \frac{1}{2}(1 - \epsilon)\beta^{-1}\nu ;$$

this follows from the continuity of f and the fact that $f(\bar{z}, \bar{p}) = 0$. Now let $\tilde{z} \in N_\epsilon$ and $\tilde{p} \in M_\epsilon$, and consider the function Φ defined by

$$\Phi(z) := z - f_1'(\bar{z}, \bar{p})^{-1} f(z, \tilde{p}) .$$

For any $z \in \bar{B}(\tilde{z}, \frac{1}{2}\nu)$ (the closure of $B(\tilde{z}, \frac{1}{2}\nu)$), we have since $\bar{B}(\tilde{z}, \frac{1}{2}\nu) \subset B(\bar{z}, \nu)$,

$$\Phi'(z) = I - f_1'(\bar{z}, \bar{p})^{-1} f_1'(z, \tilde{p})$$

$$= f_1'(\bar{z}, \bar{p})^{-1}[f_1'(\bar{z}, \bar{p}) - f_1'(z, \tilde{p})] ;$$

thus

$$\|\Phi'(z)\| \leq \beta \|f_1'(\bar{z}, \bar{p}) - f_1'(z, \tilde{p})\|$$

$$\leq \beta(\epsilon\, \beta^{-1}) = \epsilon ,$$

so that Φ is a contraction on $\bar{B}(\tilde{z}, \frac{1}{2}\nu)$. Also,

$$\|\Phi(\tilde{z}) - \tilde{z}\| \leq \|f_1'(\bar{z}, \bar{p}^{-1}\| \; \|f(\tilde{z}, \tilde{p})\|$$

$$\leq \beta[\tfrac{1}{2}(1 - \epsilon)\beta^{-1}\nu] = (1 - \epsilon)(\tfrac{1}{2}\nu) .$$

By the contraction mapping theorem [8, Th. 1 (1. XVI)], there is then some $z' \in \bar{B}(\tilde{z}, \frac{1}{2}\nu)$ such that $z' = \Phi(z')$ and

$$\|z' - \tilde{z}\| \leq (1 - \epsilon)^{-1} \|\Phi(\tilde{z}) - \tilde{z}\|$$

$$\leq (1 - \epsilon)^{-1}\beta \|f(\tilde{z}, \tilde{p})\| .$$

However, $\Phi(z') = z'$ implies $f(z', \tilde{p}) = 0$, and since $z' \in \bar{B}(\tilde{z}, \frac{1}{2}\nu) \subset B(\bar{z}, \nu) \subset N$, we have $z' = Z(\tilde{p})$. Hence

$$\|\tilde{z} - Z(\tilde{p})\| \leq (1 - \epsilon)^{-1}\beta \|f(\tilde{z}, \tilde{p})\|,$$

which completes the proof. We remark that the contraction technique used in this proof is essentially that of [8, Th. 3 (1. XVI)].

Note that if in Theorem 2 we had taken $\tilde{z} = Z(\hat{p})$ for some $\hat{p} \in M$ with $Z(\hat{p}) \in N_\epsilon$, we should have obtained

$$\|Z(\hat{p}) - Z(\tilde{p})\| \leq (1 - \epsilon)^{-1}\beta \|f(Z(\hat{p}), \hat{p}) - f(Z(\hat{p}), \tilde{p})\|,$$

since $f(Z(\hat{p}), \hat{p}) = 0$. This implies that the behavior of the Kuhn-Tucker triple $Z(p)$ for p near $\bar{p}$ may be determined by studying the behavior of f as its second argument varies. We use this fact in the next section to derive asymptotic convergence rates for a class of nonlinear-programming algorithms.

3. CONVERGENCE RATES FOR A CLASS OF ALGORITHMS. Several algorithms for nonlinear programming, including the constrained Newton method of Levitin and Polyak [9], the reverse-convex programming algorithms of Rosen [16] and of Meyer [11, 12], the method of Rosen and Kreuser [17], the algorithm of Wilson [18], and the method given by the author in [14], can be formulated as particular cases of the following procedure, which we call the general recursive algorithm after the "recursive programs" formulated and studied by Day [3, 4]. In this case, we let $P = R^{n+m+q}$ in Theorem 1, so that the perturbations lie in the (x, u, v)-space. We assume that the functions $\theta(x, p)$, $g(x, p)$ and $h(x, p)$ are given; these will, of course, vary with the different concrete realizations of the algorithm. The method can be stated as follows:

1. Start with $z_0 := (x_0, u_0, v_0)$; set $k := 0$.

2. Having z_k, let z_{k+1} be a Kuhn-Tucker triple of the program $(1\{z_k\})$; if there is more than one such triple, choose z_{k+1} to be closest in norm to z_k among all such triples (note that z_{k+1} need not be uniquely defined by this procedure).

3. Test z_{k+1} for convergence; either stop or set $k := k+1$ and go to step 2.

Note that this formulation is general enough to cover many one-point iterative algorithms: that is, algorithms which use only information at the current point, as opposed to trial and error, memory, random searches, etc.

The idea which we shall use to analyze the behavior of this algorithm is basically quite simple. Consider the first step of the algorithm: if z_0 is close to $\bar{z}$, then since z_1 is the Kuhn-Tucker triple of $(1\{z_0\})$ closest to z_0, it must in fact be $Z(z_0)$. Similarly, if we can show that z_k is close enough to $\bar{z}$, then we must have $z_{k+1} = Z(z_k)$. It follows that the general recursive algorithm amounts to a simple successive-substitution iteration on the implicit function Z. Obviously, we do not have an explicit expression for Z. However, to analyze this type of iteration we do not really need such an expression; we require only certain bounds on the variation of Z. Recall that for z near $\bar{z}$ we have from Theorem 2 (with $\tilde{z} = \tilde{p} = Z(z)$) the bound

$$\|Z^2(z) - Z(z)\| \leq (1-\epsilon)^{-1} \|f_1'(\bar{z}, \bar{z})^{-1}\| \, \|f(Z(z), Z(z)) - f(Z(z), z)\|,$$

where $Z^2(z)$ denotes the composition $Z(Z(z))$. If we now impose upon the algorithm the conditions that for some $\alpha \in \mathbb{R}$ and for each z in a given neighborhood of $\bar{z}$ we have

$$\|f(Z(z), Z(z)) - f(Z(z), z)\| \leq \alpha \|Z(z) - z\|^{\lambda}, \quad \lambda \geq 1, \tag{H1}$$

then we obtain

$$\|Z^2(z) - Z(z)\| \leq \gamma_\epsilon \|Z(z) - z\|^{\lambda},$$

where $\gamma_\epsilon := (1-\epsilon)^{-1} \alpha \|f_1'(\bar{z}, \bar{z})^{-1}\|$, and this bound permits us to prove the existence and convergence of $\{z_k\}$. To identify the limit of this sequence with a Kuhn-Tucker triple of (1), it is necessary to require some relationship between the problems (1) and $(1\{z\})$. Appropriate conditions to ensure this relationship are that

$$f(z) = f(z, z) \text{ and } f'(z) = f_1'(z, z), \tag{H2}$$

for each z in some open neighborhood of $\bar{z}$, where

$$f(z) := [\mathcal{L}_1'(x,u,v),\ u_1 g_1(x), \ldots, u_m g_m(x),\ h_1(x), \ldots, h_q(x)]^T$$

is defined for (1) in the same way as $f(z, p)$ was defined for (1{p}). Note that (H2) is required to hold only for $p = z$; its verification for a particular algorithm is usually a very simple matter.

We now state these convergence results in precise form in the following theorem.

THEOREM 3: Let $\bar{z}$ be a Kuhn-Tucker triple of (1) at which the second-order sufficiency conditions are satisfied with strict complementary slackness and linear independence of the gradients to the active constraints. Suppose that the general recursive algorithm is applied to (1) in a form for which the functions $\theta(x, z)$, $g(x, z)$ and $h(x, z)$ satisfy the differentiability and continuity hypotheses of Theorem 1, and that both (H1) and (H2) are satisfied in some open neighborhood of $\bar{z}$. Then the following results hold:

a. (Linear Convergence): If $\lambda = 1$ and $\zeta := \alpha \|f_1'(\bar{z}, \bar{z})^{-1}\| < 1$, then there is an open neighborhood $W_1(\bar{z})$ such that if the algorithm is started at any $z_0 \in W_1$, the sequence $\{z_k\}$ exists and converges to $\bar{z}$, with

$$\|\bar{z} - z_k\| \leq 2(1 - \zeta)^{-1} \|z_1 - z_0\| [\tfrac{1}{2}(1 + \zeta)]^k$$

for each $k \geq 0$.

b. (Superlinear Convergence): If $\lambda > 1$, then there exists an open neighborhood $W_\lambda(\bar{z})$ such that if the algorithm is started at any $z_0 \in W_\lambda$, the sequence $\{z_k\}$ exists and converges to $\bar{z}$, with R-order at least λ; specifically, for all $k \geq 0$ we have

$$\|\bar{z} - z_k\| \leq \mu\sigma^{(\lambda^k)},$$

where $\sigma = (2\zeta)^{1/(\lambda-1)} \|z_1 - z_0\| < 1$ and $\mu = (2\zeta)^{1/(1-\lambda)}(1-\sigma^{\lambda-1})^{-1}$.

PROOF: We shall first show that $\bar{z}$ is a Kuhn-Tucker triple of $(1\{\bar{z}\})$ which satisfies the hypotheses of Theorem 1, and then apply Theorem 2 together with (H1) and known results from numerical analysis to investigate the existence and behavior of $\{z_k\}$.

The assumption that $f(z) = f(z, z)$ means, for $z = \bar{z}$, that $\bar{z}$ satisfies the equalities of the Kuhn-Tucker conditions for $(1\{\bar{z}\})$. The equality of $f'(\bar{z})$ and $f_1'(\bar{z}, \bar{z})$ implies that $g(\bar{x}) = g(\bar{x}, \bar{z})$, $g'(\bar{x}) = g_1'(\bar{x}, \bar{z})$, $h'(\bar{x}) = h_1'(\bar{x}, \bar{z})$, and $\mathfrak{L}_{11}''(\bar{x}, \bar{u}, \bar{v}) = \mathfrak{L}_{11}''(\bar{x}, \bar{u}, \bar{v}, \bar{z})$. The first equation shows that $g(\bar{x}, \bar{z}) \leq 0$, and we know already that $\bar{u} \geq 0$, so $\bar{z}$ is a Kuhn-Tucker triple for $(1\{\bar{z}\})$; the last three equations permit us to conclude that the second-order sufficiency conditions hold there. Strict complementary slackness follows from the equality of $g(\bar{x})$ and $g(\bar{x}, \bar{z})$ and linear independence of the gradients to the active constraints is immediate. The conclusions of Theorems 1 and 2 are therefore valid with $\bar{p} = \bar{z}$, and we may with no loss of generality take $M(\bar{z})$ to be contained in the open neighborhood of $\bar{z}$ in which (H1) and (H2) hold. Let $B(\bar{z}, \eta)$ be an open ball contained in N, and let $\bar{B}(\bar{z}, \rho_\epsilon)$ be a closed ball contained in M, with $\rho_\epsilon \leq \frac{1}{3}\eta$ and such that for any $z \in \bar{B}(\bar{z}, \rho_\epsilon)$ we have $Z(z) \in M_\epsilon \cap N_\epsilon \cap B(\bar{z}, \frac{1}{3}\eta)$, where M_ϵ and N_ϵ are the neighborhoods in Theorem 2; of course they, and hence also ρ_ϵ, depend on ϵ. Using the fact that $\bar{z}$ is an isolated Kuhn-Tucker triple of (1) (because $f'(\bar{z})$ is nonsingular; see [10, p. 231]), we may also assume that ρ_ϵ has been chosen to be so small that there is no Kuhn-Tucker triple of (1), other than $\bar{z}$, in $B(\bar{z}, \rho_\epsilon)$. Applying Theorem 2 with $\tilde{p} = \tilde{z} = Z(z)$, we have that for $z \in \bar{B}(\bar{z}, \rho_\epsilon)$,

$$\|Z^2(z) - Z(z)\| \leq (1-\epsilon)^{-1} \|f_1'(\bar{z}, \bar{z})^{-1}\| \, \|f(Z(z), Z(z)) - f(Z(z), z)\| .$$

Invoking (H1), we find that for some fixed $\lambda \geq 1$ and each $z \in \bar{B}(\bar{z}, \rho_\epsilon)$,

$$\|Z^2(z) - Z(z)\| \leq \gamma_\epsilon \|Z(z) - z\|^\lambda .$$

Finally, if $z \in B(\bar{z}, \rho_\epsilon)$ then both z and $Z(z)$ belong to $B(\bar{z}, \frac{1}{3}\eta)$, so $\|Z(z) - z\| < 2\eta/3$. But any Kuhn-Tucker triple $\hat{z}$ of $(1\{z\})$, other than $Z(z)$, must lie outside N, so we must have $\|\hat{z} - z\| > 2\eta/3$. Thus $Z(z)$ is the unique Kuhn-Tucker triple of $(1\{z\})$ closest to z; hence if the general recursive algorithm is applied to any $z_k \in B(\bar{z}, \rho_\epsilon)$ we shall have $z_{k+1} = Z(z_k)$.

Now let $\varphi_\epsilon(t) := \gamma_\epsilon t^\lambda$ for $t \geq 0$, and denote by $\varphi_\epsilon^n(t)$, $n \geq 1$, the n-fold composition of φ_ϵ; $\varphi_\epsilon^0(t) := t$. The function φ_ϵ is a simple kind of

nonlinear majorant [13, §12.4] for the iterative process we are considering, and it will aid us in the analysis. For $t_0 = 0$ and some fixed $t_1 \geq 0$, we consider the scalar iterative process

$$t_{k+1} - t_k := \varphi_\epsilon(t_k - t_{k-1}), \quad k = 1, 2, \ldots .$$

The behavior of this process will enable us to draw conclusions about the behavior of the sequence $\{z_k\}$. In fact, it follows from [13, Ths. 12.4.3, 12.4.4] that if $t_1 \geq \|Z(z_0) - z_0\|$ and if $t_k \to t_*$ with $\|\bar{z} - z_0\| + t_* < \rho_\epsilon$, then (1) all the z_k lie in $\bar{B}(\bar{z}, \rho_\epsilon)$, so that the sequence $\{z_k\}$ is well defined, (2) the z_k converge to some z_* with $\|z_* - z_k\| \leq t_* - t_k$ for each k, and (3) we have $Z(z_*) = z_*$. Hence z_* is a Kuhn-Tucker triple of $(l\{z_*\})$, and applying (H2) we see that it is also a Kuhn-Tucker triple of (l); but $z_* \in B(\bar{z}, \rho_\epsilon)$, and from the local uniqueness of $\bar{z}$ it follows that we must have $z_* = \bar{z}$.

We note that $t_* = \sum_{i=0}^{\infty} \varphi_\epsilon^i(t_1 - t_0)$, and that t_* is a continuous, increasing function of t_1 on any interval $[0, T]$ for which $\gamma_\epsilon T^{\lambda-1} < 1$; we may write $t_* = t_*(t_1)$ for emphasis. If we now choose $W_\lambda(\bar{z})$ to be the open neighborhood $\{z \in B(\bar{z}, \rho_\epsilon) \mid \gamma_\epsilon \|Z(z) - z\|^{\lambda-1} < 1, \ \|\bar{z} - z\| + t_*(\|Z(z) - z\|) < \rho_\epsilon\}$, then with $z_0 \in W_\lambda(\bar{z})$ and $t_1 := \|Z(z_0) - z_0\|$ we see that the conditions of the previous paragraph are satisfied.

Now, if $\lambda = 1$ and $\zeta < 1$, we choose $\epsilon := (1-\zeta)/(1+\zeta)$, so that $\gamma_\epsilon = \frac{1}{2}(1+\zeta) < 1$. Then since $\varphi_\epsilon^n(t_1 - t_0) = \gamma_\epsilon^n t_1$, we find that $t_k = t_1 \sum_{i=0}^{k-1} \gamma_\epsilon^i$, so $t_* = (1-\gamma_\epsilon)^{-1} t_1$. In this case, the error bound is

$$\|\bar{z} - z_k\| \leq t_* - t_k = t_1[(1-\gamma_\epsilon)^{-1} - (1-\gamma_\epsilon^k)(1-\gamma_\epsilon)^{-1}] = 2(1-\zeta)^{-1}\|z_1 - z_0\|[\tfrac{1}{2}(1+\zeta)]^k,$$

so the result of part (a) holds.

For part (b.), assume $\lambda > 1$ and choose $\epsilon = 1/2$. The error bound becomes

$$\|\bar{z} - z_k\| \leq t_* - t_k = \sum_{i=k}^{\infty} \varphi_\epsilon^i(\|z_1 - z_0\|) =$$

$$= \gamma_\epsilon^{1/(1-\lambda)} \sum_{i=k}^{\infty} \sigma^{(\lambda^i)} = \gamma_\epsilon^{1/(1-\lambda)} \sigma^{(\lambda^k)} \sum_{i=0}^{\infty} \sigma^{(\lambda^i - 1)\lambda^k}$$

where $\sigma := \gamma_\epsilon^{1/(\lambda-1)} \|z_1 - z_0\| < 1$. But using $\lambda^i - 1 \geq i(\lambda-1)$, we have

$$\sum_{i=0}^{\infty} \sigma^{(\lambda^i-1)\lambda^k} \leq \sum_{i=0}^{\infty} \sigma^{i(\lambda-1)\lambda^k} \leq (1-\sigma^{\lambda-1})^{-1} ,$$

so

$$\|\bar{z} - z_k\| \leq \gamma_\epsilon^{1/(1-\lambda)} (1-\sigma^{\lambda-1})^{-1} \sigma^{(\lambda^k)} ,$$

which completes the proof.

4. EXAMPLES. We give here the forms of θ, g and h, and the values of λ, for three concrete realizations of the general recursive algorithm.

a. Constrained Newton Method (Levitin-Polyak [9]):

$$\theta(x, z_k) = \theta(x_k) + \theta'(x_k)(x - x_k) + \frac{1}{2}\theta''(x_k)(x - x_k)^2$$

$$g(x, z_k) = g(x)$$

$$h(x, z_k) = h(x)$$

$$\lambda = 2 .$$

b. Wilson's Method (Wilson [18]):

$$\theta(x, z_k) = \theta(x_k) + \theta'(x_k)(x - x_k) + \frac{1}{2}\theta''(x_k)(x - x_k)^2 + \frac{1}{2}\sum_{i=1}^{m} (u_k)_i g_i''(x_k)(x - x_k)^2 + \frac{1}{2}\sum_{j=1}^{q} (v_k)_j h_j''(x_k)(x - x_k)^2$$

$$g(x, z_k) = Lg(x_k, x) := g(x_k) + g'(x_k)(x - x_k)$$

$$h(x, z_k) = Lh(x_k, x) := h(x_k) + h'(x_k)(x - x_k)$$

$$\lambda = 2 .$$

c. Method of [14]:

$$\theta(x, z_k) = \theta(x) + u_k^T[g(x) - Lg(x_k, x)] + v_k^T[h(x) - Lh(x_k, x)]$$

$$g(x, z_k) = Lg(x_k, x)$$

$$h(x, z_k) = Lh(x_k, x)$$

$$\lambda = 2 .$$

A consequence of the analysis given above is that if θ, g and h have the smoothness property required in Theorem 1, and if $\bar{z}$ is a Kuhn-Tucker triple of (1) satisfying the second-order sufficiency conditions, strict complementary slackness, and linear independence of the gradients to the active constraints, then $\bar{z}$ is a point of attraction for any of the three algorithms specified above; further, each algorithm is R-quadratically convergent in the sense of [13]. To illustrate how easy it is to prove this using Theorem 3, we shall go through the verification of (H1) for the method of [14]. In fact, we shall establish the stronger result that for any z_a, z_b near $\bar{z}$, we have

$$\|f(z_a, z_a) - f(z_a, z_b)\| \leq \alpha \|z_a - z_b\|^2 . \tag{3}$$

From the specification of the algorithm, we find that

$$f(z_a, z_a) = \begin{bmatrix} \theta'(x_a) + u_a^T g'(x_a) + v_a^T h'(x_a) \\ (u_a)_1 g_1(x_a) \\ \vdots \\ (u_a)_m g_m(x_a) \\ h_1(x_a) \\ \vdots \\ h_q(x_a) \end{bmatrix} = f(z_a) ,$$

while

$$f(z_a, z_b) = \begin{bmatrix} \theta'(x_a) + u_b^T[g'(x_a) - g'(x_b)] + v_b^T[h'(x_a) - h'(x_b)] \\ \quad + u_a^T g'(x_b) + v_a^T h'(x_b) \\ (u_a)_1 Lg_1(x_a, x_b) \\ \vdots \\ (u_a)_m Lg_m(x_a, x_b) \\ Lh_1(x_a, x_b) \\ \vdots \\ Lh_q(x_a, x_b) \end{bmatrix}$$

Subtracting, we obtain

$$f(z_a, z_a) - f(z_a, z_b) = \begin{bmatrix} (u_a - u_b)^T[g'(x_a) - g'(x_b)] \\ (u_a)_1[g_1(x_a) - Lg_1(x_a, x_b)] \\ \vdots \\ (u_a)_m[g_m(x_a) - Lg_m(x_a, x_b)] \\ h_1(x_a) - Lh_1(x_a, x_b) \\ \vdots \\ h_q(x_a) - Lh_q(x_a, x_b) \end{bmatrix} \tag{4}$$

The expression on the right-hand side of (4) is clearly of second order in $(z_a - z_b)$, so in a neighborhood of $\bar{z}$ the bound in (3) will be valid with $\lambda = 2$. The verification of (H2) for this method is easy, and these two simple steps are all that is necessary to infer from Theorem 3 the implementability, convergence, and quadratic rate of convergence of this algorithm.

The results of Theorem 3 provide a framework for analyzing the local behavior of a class of methods of a certain recursive type. The nature of the hypotheses required for the application of this theorem shows clearly that the convergence rate of the general recursive algorithm does not depend directly upon how well the individual objective and constraint functions are approximated by the modified functions used in the algorithm, but rather upon how well the function f, constructed from θ, g, h and their derivatives as well as from u and v, is approximated near the Kuhn-Tucker triple $\bar{z}$. It would thus appear that if new recursive algorithms are constructed in such a way as to yield good approximations of f near $\bar{z}$, they can be expected to possess favorable rates of convergence.

The general approach taken in Section 3 was to show that the general (one-point) recursive algorithm amounted to a successive-substitution iteration on the implicit function Z, that the variation of Z near $\bar{z}$ could be bounded in terms of functions appearing explicitly in the algorithm, and that these bounds could then be used to infer the existence, convergence and convergence rate of the sequence of approximate solutions $\{z_k\}$. It is evident that nothing in this approach inherently limits its use to the class of one-point methods; this was simply the most convenient class to use in illustrating the general approach. One could just as easily consider two-point or multipoint methods for the solution of nonlinear programs, analogous to those analyzed extensively in [13] for solving systems of nonlinear equations. Analyses for these methods, paralleling that of Section 3 for one-point algorithms, could be constructed by selecting different concrete realizations of the perturbation space P appearing in Theorems 1 and 2. It would be useful and interesting to have computational experience available for some of these methods.

REFERENCES

[1] C. Berge, Topological spaces (Macmillan, New York, 1963).

[2] G. B. Dantzig, J. Folkman and N. Z. Shapiro, "On the continuity of the minimum set of a continuous function," J. Math. Anal. Appl. 17 (1967), 519-548.

[3] R. H. Day, Recursive programming and production response (North-Holland, Amsterdam, 1963).

[4] R. H. Day and [illegible] Kennedy, "Recursive decision systems: an existence analysis," Econometrica 38 (1970) 666-681.

[5] J. P. Evans and F. J. Gould, "Stability in nonlinear programming," Operations Res. 18 (1970) 107-118.

[6] A. V. Fiacco, "Sensitivity analysis for nonlinear programming using penalty methods," Technical Paper, Serial T-275; Institute for Management Science and Engineering, The George Washington University, Washington, D. C., March 1973. See also R. L. Armacost and A. V. Fiacco, "Computational experience in sensitivity analysis for nonlinear programming," Technical Paper, Serial T-276; Institute for Management Science and Engineering, The George Washington University, Washington, D. C., February 1973, forthcoming in Math. Programming.

[7] A. V. Fiacco and G. P. McCormick, Nonlinear programming: sequential unconstrained minimization techniques (Wiley, New York 1968).

[8] L. V. Kantorovich and G. P. Akilov, Functional analysis in normed spaces (Macmillan, New York 1964).

[9] E. S. Levitin and B. T. Polyak, "Constrained minimization methods," U.S.S.R. Comput. Math. Math. Phys. 6, 5 (1966) 1-50 [Original in Russian: Zh. vychisl. Mat. mat. Fiz. 6, 5 (1966) 787-823].

[10] G. P. McCormick, "Penalty function versus nonpenalty function methods for constrained nonlinear programming problems," Math. Programming 1 (1971) 217-238.

[11] R. R. Meyer, "The solution of non-convex optimization problems by iterative convex programming," dissertation. University of Wisconsin, Madison, 1968.

[12] R. R. Meyer, "The validity of a family of optimization methods," SIAM J. Control 8 (1970) 41-54.

[13] J. M. Ortega and W. C. Rheinboldt, Iterative solution of nonlinear equations in several variables (Academic Press, New York 1970).

[14] S. M. Robinson, "A quadratically-convergent algorithm for general nonlinear programming problems," Math. Programming 3 (1972) 145-156.

[15] S. M. Robinson and R. H. Day, "A sufficient condition for continuity of optimal sets in mathematical programming," J. Math. Anal. Appl. 45 (1974) 506-511.

[16] J. B. Rosen, "Iterative solution of nonlinear optimal control problems," SIAM J. Control 4 (1966) 233-244.

[17] J. B. Rosen and J. Kreuser, "A gradient projection algorithm for nonlinear constraints," in Numerical Methods for Nonlinear Optimization, Ed. F. A. Lootsma (Academic Press, New York, 1972), pp. 297-300.

[18] R. B. Wilson, "A simplicial algorithm for concave programming," dissertation. Graduate School of Business Administration, Harvard University, Cambridge, 1963.

MODELING AND SIMULATION OF CELLULOSE/Tv CELLULASE HYDROLYSIS

Chul Kim

Pioneering Research Laboratory
U.S. Army Natick Laboratories
Natick, Massachusetts 01760

ABSTRACT

The kinetics of the hydrolysis of cellulose by Tv-cellulase was investigated. A kinetic scheme based on experimental observations and theoretical hypotheses was proposed and simulated by using IBM S/360 CSMP and MIMIC simulators. Due to the complicated nature of the cellulose/cellulase system, a simple Michaelis-Menton type kinetics did not apply and the usually sought steady state solution was found inadequate. A user supplied subprogram and an IBM S.S.P. were of help to find the best estimates of the system parameters.

The results in comparison to the experimental data were satisfactory.

MATHEMATICAL MODELING

Developing the models of chemical kinetics consists of solving the following problems:

(1) formulation of reaction pathways and deriving kinetic rate equations.
(2) testing the significance of the postulated model variables and reaction steps to find the maximum number of variables and/or reaction steps.

Before the rate of a reaction can be meaningfully discussed, the reaction system must be defined as precise as possible. In practice, experiments involve the determination of an average rate of reaction of a large number of molecules resulting in proposed pathways or mechanisms which usually present a sequence of reaction steps. Starting with simple experimental results (e.g., end products or reactants reaction rates), a diverse experimental design and reliable analysis techniques are desirable to provide maximum information. Once the information is available, models can be established. In many cases more than one reaction mechanism can describe the observed experimental results. Screening and testing are required of the models for validity of each element and reaction steps to support the experimental observations (Fig 1).

The rate constants and system parameters must be determined for the kinetic model thus proposed. Many methods have been suggested (1) to obtain the best estimates of the constants from experimental data.

In general the more complex the mathematical model description of a system is, the more difficult is its handling to obtain the solution. Therefore, the art of modeling is based on the policy to treat the system in just enough detail so that the solution will provide information about the essential features of its behavior to within desired accuracy. However, the model should retain certain process elements that are physiolcgically reasonable. This often results in a high degree of complexity which is inevitable. The analyst thus, should carefully diagnose the problem and represent it by a proper compromise between the necessary detail, the availability of experimental information and the required mathematical tools.

In making simplifying assumptions which are remote from the physics of the process, the experimental observations or the mathematical compatibility, purely to make the mathematics tractable, there always exists danger to overlook certain important aspects. Also there is little merit to attempt an analytical solution, however elegant it may be, if there is a high degree of complexity so that it is difficult to obtain numbers from the solution.

Cellulose/Cellulase System

(a) Mass transfer limitations

The cellulose/cellulase biochemical system is very complex because cellulose is an insoluble polymer which contains a range of substrate varying from amorphous and reactive to crystalline and highly resistant parts and cellulase is a mixture of several enzyme components. It forms a mixture of solid cellulose particles suspended in liquid enzyme solution. The enzyme is catalytic. Unlike many heterogeneous catalytic systems, the enzyme (catalyst) molecules migrate to the cellulose (reactant) and the product released after digestion.

Due to the heterogeneity, it is conceivable that various mass transfer resistances may play significant roles in overall reaction rate. The bulk phase and film mass transfer rates depend on the size of cellulose particles, the cellulose concentration and the degree of agitation or the Reynolds' number of the mixture being stirred. Experiments have been carried out to study the mass transfer limitations. When the agitation speed exceeded 100 r.p.m. in a batch reactor using pure cellulose (SFBW 200) as a substrate with concentration levels of 2 $\sim$ 10 wt. %, it was observed that the mass transfer resistances were negligible. This is an indication that the bulk and film resistances can be made negligible with proper experimental conditions. Thus, with considerable size reduction of cellulose and an adequate mixing so that the cellulose particles are maintained in good suspension in the slurry mixture, it is safe to assume that there is negligible enzyme concentration gradient in the solution. This considerably simplifies the mathematical handling. However, further investigation may be proper since there is energy demand

to meet the forementioned experimental conditions. No experiments were conducted regarding the pore diffusion of enzyme, however, it is assumed that the pore diffusion resistance is likewise insignificant due to the macromolecular nature of enzyme molecules.

(b) Cellulase Modes of Action

The adsorption of cellulase in contact with the cellulose particles provides the only mass transfer resistance and it is important to establish a relation which describes the adsorption characteristics. Since the cellulose and the cellulase consist of more than one component, the modes of action of cellulase on different portions of cellulose are of fundamental significance in the elucidation of cellulase adsorption and the understanding of overall kinetic process.

At present there are two distinct theoretical postulations, attributed largely to Reese and Mandels (2) and Wood *et al.* (3) to explain the modes of action (Fig 2).

Postulate 1:

C_1 is an enzyme that reduces bonds between cellulase chains by opening up the crystalline structure to convert the crystalline cellulose to amorphous and/or reactive cellulose. C_X (endo- and exo-glucanases) hydrolyzes the more susceptible amorphous and/or reactive cellulose by removing glucose units endwise from the nonreducing ends (exo-) and by primarily random fission of longer chain length (endo-).

Postulate 2:

C_X acts on the crystalline cellulose to generate free ends which are more susceptible to enzymic attack and C_1 is an enzyme which hydrolyzes the reactive ends of the cellulose produced by C_X action.

The postulated theories are based on the specific experimental observations and offer similar qualitative explanations for the separate and distinctive actions of the cellulase components. In either case, it is an essential requirement that both components are needed in order to achieve saccharification of cellulose material to a significant extent. The catalytic actions by these components are synergistic. The rate of degradation of the crystalline cellulose is shown to be very slow in comparison to the hydrolysis rate of the amorphous and/or reactive cellulose. It is thus, difficult to determine a meaningful concentration ratio of C_1 to C_X or vice versa. The postulations are not affirmative and leave room for further intensive investigation.

Current study assumes somewhat different enzymic action. The solution enzyme is considered as a single component enzyme and distinctive catalytic actions appear only

when enzyme is adsorbed on different portions of cellulose matrix. The crystalline bonds breaking enzyme is postulated as having $C_1 + C_x$ complex form and C_1 or C_x separately acts on the hydrolysis step to produce the reducing sugars. This scheme not only maintains basic similarity to above-mentioned theories, but also eliminates difficulties in determining the concentrations of C_1 and C_x separately. The initial conditions for the cellulase can readily be defined in terms of total enzyme concentration in the rate equations.

(c) Cellulase Adsorption

Cellulase is strongly adsorbed by cellulose. The amount of cellulase adsorbed depends on the available sorption site which is a function of cellulose particle size and concentration of cellulose at fixed experimental conditions (50°C, pH 4.8). The initial adsorption is fast and the adsorption is continued at a slower rate for a short time period. The rapid initial uptake is due to high cellulose-cellulase concentration ratio. More than 90% of initial adsorption took place by 10% cellulose (SFBW 200) at prescribed experimental conditions in a batch reactor. Before an appreciable production of glucose (initial 3 $\sim$ 4 hrs) the adsorption continuously takes place until the cellulase concentration reaches a state of "pseudosaturation". Once the cellulose is pseudosaturated with cellulase, negligible amount of uptake is observed, thus bulk solution cellulase concentration remaining almost at a constant level. The cellulase adsorbed forms cellulose-cellulase complex and digestion starts. As digestion continues to produce glucose on a conversion level of 40%, the cellulase concentration in the solution increases indicating the release of cellulase from cellulose. Typical digestion curves are shown in Fig (3). After an hour of reaction, 15% volume of sample was taken and supernate separated from the centrifuged sample. Added with fresh buffer solution into precipitate and let the reaction continue in an shaker-incubator. As shown (Fig 3B), glucose production rate is nearly equal as in continued reaction in the original batch reactor (Fig 3A). A slight lagging of curve B compared with curve A is believed to be due to continued adsorption for a short initial time. At about 8 hrs of reaction curve B overtakes curve A indicating a possible effect of product inhibition. These data were reproducible leading to a conclusion that most of the adsorbed enzyme is held by cellulose and is primarily responsible for the cellulose degradation. The enzyme is released when digestion is continued for a prolonged time and the cellulose is depleted.

(d) Product Inhibition

The hydrolysis takes place fast for the initial time period (1 $\sim$ 8 hrs) and levels off considerably for a prolonged reaction time (Fig 3). This is believed to be due to the inhibitory action by product or changes of cellulose susceptibility or both. In modeling standpoint the effect of susceptibility change can be depicted in the reaction step for the degradation of the crystalline cellulose. The retardness of the hydrolysis rate which

becomes apparent when the product concentration builds up is mainly attributable to the inhibition by the reaction product. The most common type of product inhibition, i.e., a reaction between the solution enzyme and the product is not likely to affect the productivity since this system is heterogeneous and the enzyme adsorbed on the cellulose matrix for the short initial time period is primarily responsible for the reaction. The most probable inhibitory step is due to side reactions between the enzyme-substrate complexes and the product. These reactions usually are reversible processes whose equilibria maintain the resulting reaction velocities to a certain constant level at specified initial conditions.

These experimental observations along with theoretical considerations lead one to establish the modeling policy (I11 1) and a proposal kinetic scheme (Fig 4, I11 2, I11 3). This kinetic model thus, represents one of the most probable reaction path for this system.

INITIAL CONDITIONS

(a) The cellulose concentration

Before an actual simulation is performed, it is required to specify initial conditions to integrate the system differential equations. To assign initial values to each cellulose component and the cellulase is one of the difficulties encountered since there exist numerous factors by which the cellulose structure can be altered. The cellulose is pretreated for the size reduction to fine particles before it enters the digestion vessel. Experimental observation shows significant differences in the reactivities of cellulose which are pretreated by various mechanical means under different physical conditions (4).

Even though it appears rather crude the current method to determine the composition (the ratio of crystalline to noncrystalline) is by hydrolyzing the specific substrate sample by P.w.* cellulase culture for an extended long time period. P.w. cellulase contains an enzyme component capable of digesting noncrystalline part of cellulose only. Approximately 12% of the cellulose used in this study (SFBW 200, 200 mesh) can be hydrolyzed by P.w. cellulase in 48 hrs and thereafter almost negligible digestion takes place for prolonged time. This indicates only 12% of total SFBW 200 cellulose is of readily reactive form leaving 88% crystalline, resistant part.

At present there is not any decisive way to estimate the compositions of various forms of cellulose thus forcing the use of forementioned experimental determinations.

It is enhanced, however, that further investigation to be carried out in regard to the development of any deterministic relations between the cellulose compositions and various physical and mechanical factors involved in processes of cellulose pretreatment.

*Pestalotiopsis westerdijkii

Added significance for the extended study in this regard can be attributed to the fact that there are energy and cost demands in these processes to increase the cellulose accessibility and reactivity.

(b) The Enzyme Concentration (Activity or Strength)

The interesting and important feature of adsorbed enzymes is their mixed function catalytic action. The bulk solution enzyme is homogeneous, while the catalytic action of enzyme adsorbed on the substrate matrix is heterogeneous.

In cellulose/cellulase system the catalytic action of adsorbed enzyme is of particular significance, for the enzyme adsorbed in initial short period of reaction appears to be primarily responsible in its catalytic action. The requirement of information on the adsorbed enzyme activity is essential in the kinetics study of this system.

Currently utilized information on cellulase activity and concentration provides:

1) Total Protein Content (Pr)
2) Filter Paper Activity (FPA)
3) IUB Unit

Total protein content is a physical entity that can be measured in definite quantity. It is algebraically aditive in the amount present in bulk solution and on substrate matrix. Not all the proteins possess the catalytic action of enzyme and it is difficult to make any presumption that;

a) the concentration ratio of enzyme-protein (EPr) and nonenzyme-protein (NPr) might be proportional to total protein concentration

b) regardless whether a) is true or not the ratio EPr/NPr in adsorbed state may have a defined correlation with total adsorbed protein. Neither total protein content nor total protein adsorbed thus appears to measure proper enzyme action.

The FPA determined by a standard assay procedure in this lab is in general acceptable and is being used as the "effective" catalytic action of cellulase. FPA is expressed in the amount of sugar produced from 50 mg of standard substrate (Whatman #1 filter paper) in ½ ml of enzyme preparation and 1 ml of buffer (Na citrate) mixture after an hour of "reaction" under certain optimal conditions (50°C, pH 4.8).

A typical FPA vs. dilution (or total protein content) curves are shown in Fig 5.

The enzyme preparation from QM9414 mutant of T.v. fungus was diluted from 1/10 to full strength (therefore protein concentrations were also diluted). The FPA obtained measures the catalytic action of enzyme adsorbed on the filter paper. As shown the activities of enzyme are not linearly proportional with the dilution factors.

A simple hyperbolic relation (Eq. A) between the FPA and Pr was tested and was found to fit the observed FPA vs. dilution data within the accuracy of possible experimental error. In all cases the Lineweaver-Burke type plots (Fig 6) show reasonably good linearities between the variables plotted. The significance of the constants, B_1 and B_2 are not immediately clear. The B_1 values were very close (approximately 5.0) in four different enzyme preparations when 50 mg of filter paper was used, and they differ but slightly when 100 mg and 25 mg of filter paper was used. This indicates the dependency of B_1 on the amount of substrate, hence its adsorption capacity. B_2 values, however, varied with the amount of substrate as well as with the different enzyme preparations (different batches) thus, showing the dependency on the adsorption capacity and the ratio of NPr/EPr (TABLE A, B).

In the hydrolysis system, if the initial adsorption of the total protein is determined, the corresponding FPA could be estimated.

$$FPA = \frac{B_1 P_r}{B_2 + P_r} \qquad (A)$$

Using this relation, the FPA, or the enzyme concentration can be estimated once the total protein content of the cellulase being used is known.

For the initial cellulase concentration adsorbed, Eq. (A) is modified to give

$$(FPA)_{ad} = \frac{B_1 (\alpha\ Pr)}{B_2 + (\alpha\ Pr)}$$

where $\alpha = \dfrac{\text{total protein adsorbed}}{\text{total protein}}$

SIMULATION

(a) The CSMP (I11 5) and MIMIC

Continuous system simulation languages are extremely useful tools in modeling continuous systems as well as in finding optimal parameters in the system differential equations.

In developing mathematical models of chemical reaction systems, it is well recognized that the system differential equations are large and nonlinear. One method of attacking the general problem to obtain the solutions of these rate equations as well as to determine parameters, is to program the model equations on an analogue computer and fit the generated curves to the experimental data. This could be achieved by simply changing the settings of potentiometer. The difficulties in using analogue machine are:

(1) it becomes impossible to keep track of the response to one of many parameters as the number of dependent variables and/or parameters increases.

(2) as the ranges of variations of dependent variables and/or parameters are widely spread (in complex kinetic systems this often is true), the scaling of variables into reasonable voltage levels becomes intractable.

The use of analogue computer, thus, sometimes makes it difficult to handle complicated problems in spite of easy accessibility due to its parallel nature.

On the other hand, the serial nature of the digital computer along with its lack of hardware integrator requires a skilled programming to solve these problems. The coding of integration routines to handle large and complicated system equations can also be extremely tedious and time consuming.

The simulation languages used in this investigation are digital programs which blend the best of both analogue and digital computers; the parallel nature of the analogue with the large dynamic range of digital.

With these languages the model can simply be written down either in the form of block diagrams or in the differential equations. All the variational equations are written in a structural statement form. The complexities of the integrations are carried out in the translation of the structural statements to Fortran. They are also very flexible to provide various use-oriented input, output control statements as well as to accept Fortran statements and subprograms. The ability to accept any Fortran statements and user-written Fortran programs allows the user to readily implement the use of these languages in parameter determinations.

(b) The GELG

The Gauss-Newton iterative technique is being used to determine optimal parameters in the system. The use of I.B.M. S.S.P. GELG is of great help in minimizing sequence of the least square steps and in checking convergence criterion.

(c) Parameter Determinations

The kinetic rate equations form a system of differential equations, (1) a,b.

$$\dot{C}_j = f_j\ (t, C_1, C_2\ \ldots, C_n;\ P_1,\ P_2 \ldots P_m) \qquad (j = 1,2,\ldots,J) \qquad (1)a$$

$$C_j(o) = C_{jo} \qquad (j = 1,2,\ldots,J) \qquad (1)b$$

where the C_j's are the dependent variables (concentrations, P_r's are the rate constants and system parameters, and the f_j's represent the desired functional relations of C and t.

Equation (1)b gives the known initial conditions. The rate equations are in general, nonlinear in C_j's but more often linear with respect to related parameters (rate constants). Often they are highly interacting one another thus, overall nonlinearity increases. When time appears explicity in f_i's, the system is nonautonomous.

The existence and uniqueness of the solutions of (1) are guaranteed if f_j's possess continuously uniformly bounded partial derivatives with respect to C_j's in the region of interest. This condition is usually met with the rate equations that describe any physical systems.

If C_j^i and C_{ji} are denoted as the predicted and the experimental values measured at times t_i, respectively, one criterion that can be used for the estimation of the best values of P_r's is to minimize the sum of the squares of the weighted deviations.

In the following equations the summation convention for the repeated indices is used.

The expression of the function, ∇^2, to be minimized is

$$\nabla^2 = \left[Q_{ji} \, (C_{ji} - C_j^i) \right]^2 \tag{2}$$

where Q_{ji} is the weighting factor associated with each deviation. Most frequently used weighting factors are:

(1) $Q_{ji} = 1$ equal weighting for each deviation

(2) $Q_{ji} = 1/C_{ji}$ relative deviation

(3) $$Q_{ji} = \left[\sum_{1=1}^{R} (C_{j1} - (1/R) \sum_{i=1}^{R} C_{ji})^2 \right]^{-\frac{1}{2}}$$

weighting to the varience of C_{ji}

The solution $C_j(t; P)$ can be expanded in Taylor series about the initial guesses of the parameter values,

$$C_j(t; P^*) = C_j(t; P^0) + C_{j,P_k}(t; P^0) \, P_k + C_{j,P_k}(t; P^0) \, P_k^{-2} + \cdots\cdots\cdots \tag{3}'$$

$$(k = 1, 2, \cdots\cdots, m)$$

$$(J = 1, 2, \cdots\cdots, J_o)$$

where J_o = number of the variables C_j which the experimental data are available. Eq. (3)' can be simplified by the following considerations;

(1) through the initial rate study it is possible to obtain close estimates of the constants, k_1, k_{-1}, k_2

(2) since the parameters are expressed relative to k_1, the forward reaction rate constant in the fast reaction step, the values are less than one and the rough estimates within the accuracy of the order of magnitudes can be made.

The Eq. (3)' can now be truncated to give

$$C_j(t; P^*) = C_j(t; P^0) + C_{j,P_k}(t; P^0)\,\bar{P}_k \qquad (3)$$
$$C^i_j(t_i; P^*) = C^i_j(t_i; P^0) + C^i_{j,P_k}(t_i; P^0)\,\bar{P}_k$$

where P^* = optimal parameter vector

and $\bar{P}_k = P^*_k - P^0_k$

The equations (2) and (3) are combined to obtain

$$\nabla^2 = \left[Q_{ji} \left(C_{ji} - C^i_j (P^0) - C^i_{j,P_k}(P^0)\,\bar{P}_k\right)\right]^2 \qquad (4)$$

This is a quadratic function of P_k,s with only one minimum in N-dimensional space. By setting all the first partial derivatives of ∇^2 with respect to P_k,s equal to zero, a system of algebraic equations is obtained

$$\nabla^2_{,\bar{P}_k} = 0 \qquad (k = 1, 2, \cdots\cdots, m) \qquad (5)$$

which, given equal weighting, lead to

$$\left[C_{ji} - C^i_j(P^0) \right] \; C^i_{j,P_1}(P^0) = C^i_{j,P_k}(P^0)\; C^i_{j,P_1}(P^C)\,\bar{P}_k \qquad (6)$$

The integration was performed by 4th order Runge-Kutta routine in the dynamic section of the CSMP (5) and the Gauss-Newton method was programmed to solve Eq. (6) for P_k,s in the terminal segment. Fifteen iterations were required to obtain the parameter values within the deviations of 5 to 7%.

INITIAL RATE, STEADY STATE KINETICS

The concept of steady state in chemical kinetics has been a useful notion itself was justified as a physical reality by reliable experimental techniques. Whereas the notion has also been purely conceptual without good reasoning, and used for the convenience of mathematical simplification. In complicated kinetic systems, it is virtually impossible

to obtain the analytical solution for any species of interest without simplification. Steady states assumption on certain intermediates sometimes greatly simplify the mathematical procedure by reducing the number of differential equations thus leading one to a closed form of analytical solution. Yet, not to mention of mathematically nonsensical assumptions or of the incompatibility in initial conditions, the solution obtained by the assumption of steady has its limited application. The question is then whether the criteria for time after which the steady-state hypothesis is tolerable is satisfactory.

With modern computer technology there is no doubt that more reliable solution can be obtained by using computers without the risk of inaccuracy. Approximate solution under S.S. assumption can be useful to provide a quick estimate of the system variables and/or parameters.

In Fig 8 a comparison is made between exact and approximate solutions, the initial rates (I N I) and the steady state kinetics (S S).

A comparison between predictions by the proposed model and the experimental data is shown in Fig 7 with a good agreement. For a chemical reacting system alternative kinetic schemes may sometimes serve to estimate approximate rates. It should be point out however, that a slight difference in rate estimation may often cause a considerable production cost change in an optimization sequence for a large scale plant operation.

BIBLIOGRAPHY

(1) Lapidus, L. : Chem. Eng. Progr. Symp. Ser. 57, No. 36, 126 (1961)

(2) Reese, E. T., Mandels, M. : Advances in Enzyme Hydrolysis of Cellulose and Related Materials(Reese, E. T.), Pergamon Press New York, 197 (1963)

(3) Wood, T. M., McCrae, S. I. : Biochem. J., 128, 1183 (1972)

(4) Mandels, M., Hontz, L., Nystrom, J. : 'Enzymatic Hydrolysis of Waste Cellulose', to be published

(5) IBM System/360 Continuous System Modeling Program(360A-CX-16X), User's Manual

MODELING POLICY

(1) BI-COMPOSITION OF CELLULOSE

(2) CELLULASE MODES OF ACTION

(3) NEGLIGIBLE MASS TRANSFER RESISTANCES

(4) CELLULASE ADSORPTION, COMPLEX FORMATION

(5) DECOMPOSITION OF THE COMPLEX

(6) PRODUCT INHIBITION

(7) CELLULASE DEACTIVATION

HYPOTHESES, ASSUMPTIONS

(1) CELLULOSE FIBRE IS OF LONG CYLINDRICAL FORM

(2) THE AMOUNT CELLULASE ADSORBED ON EACH PORTION OF CELLULOSE IS PROPORTIONAL TO ITS CONCENTRATION

(3) EXPONENTIAL DECAY OF THE ADSORBED CELLULASE

III 1

OVERALL KINETIC SCHEME

(A) COMPLEX FORMATION, DECOMPOSITION

$$C_A + E_A^A \underset{k-1}{\overset{k_1}{\rightleftharpoons}} X_A \xrightarrow{k_2} G_X + E_A^A$$

$$C_C + E_A^C \underset{k-3}{\overset{k_3}{\rightleftharpoons}} X_C \xrightarrow{k_4} C_A + E_A^C$$

(B) INHIBITION

$$X_A + G_X \underset{k-5}{\overset{k_5'}{\rightleftharpoons}} Y_A$$

$$X_C + G_X \underset{k-6}{\overset{k_6}{\rightleftharpoons}} Y_C$$

(C) ENZYME DEACTIVATION

$$E_A^A \xrightarrow{k_7} E_A^d$$

$$E_A^C \xrightarrow{k_7} E_A^d$$

III. 2

RATE EQUATIONS

$$\dot{C}_A = -PA + \alpha X_C$$

$$\dot{C}_C = -PC$$

$$\dot{G}_X = \dot{\gamma} X_A - QA - QC$$

$$\dot{X}_A = \gamma X_A + PA - QA$$

$$\dot{X}_C = \alpha X_C + PC - QC$$

$$\dot{Y}_A = QA$$

$$\dot{Y}_C = QC$$

where

$$PA = \frac{E_{Ao}}{C_o^{1/2}} \quad \frac{C^2_A}{C^{1/2}} \exp(-\xi\tau) - X_A / P_{k1}$$

$$PC = \beta \left[\frac{E_{Ao}}{C_o^{1/2}} \quad \frac{C^2_C}{C^{1/2}} \exp(-\xi\tau) - X_C / P_{k3} \right]$$

$$QA = \delta (X_A G_x^m - Y_A / P_{k5})$$

$$QC = \eta (X_C G_x^m - Y_C / P_{k6})$$

$$\cdot \equiv \frac{d}{d\tau} \quad \tau = k_1 t \qquad t = \text{time}$$

$$\alpha = \frac{k_4}{k_1} \quad \beta = \frac{k_3}{k_1} \quad \gamma = \frac{k_2}{k_1} \quad \delta = \frac{k_5}{k_1}$$

$$\eta = \frac{k_6}{k_1} \quad \xi = \frac{k_7}{k_1}$$

III. 3

C S M P

INITIAL

PARAM $\underline{P}^{o}$ ..... Initial parameter values

INCON C_{jo}..... Initial conditions

AFGEN C_{ji}..... Experimentel data

.

.

DYNAMIC

.

$\dot{C}_j = F_j$

$C_j = \int F_j$

.

.

TERMINAL

$\nabla^2_{,Pk} = 0$

FORTRAN SUBPROGRAMS (IBM SSP)

TIMER

PRINT $C_j, \underline{P}^{o}$

Ill 5 Continuous System Modeling Program

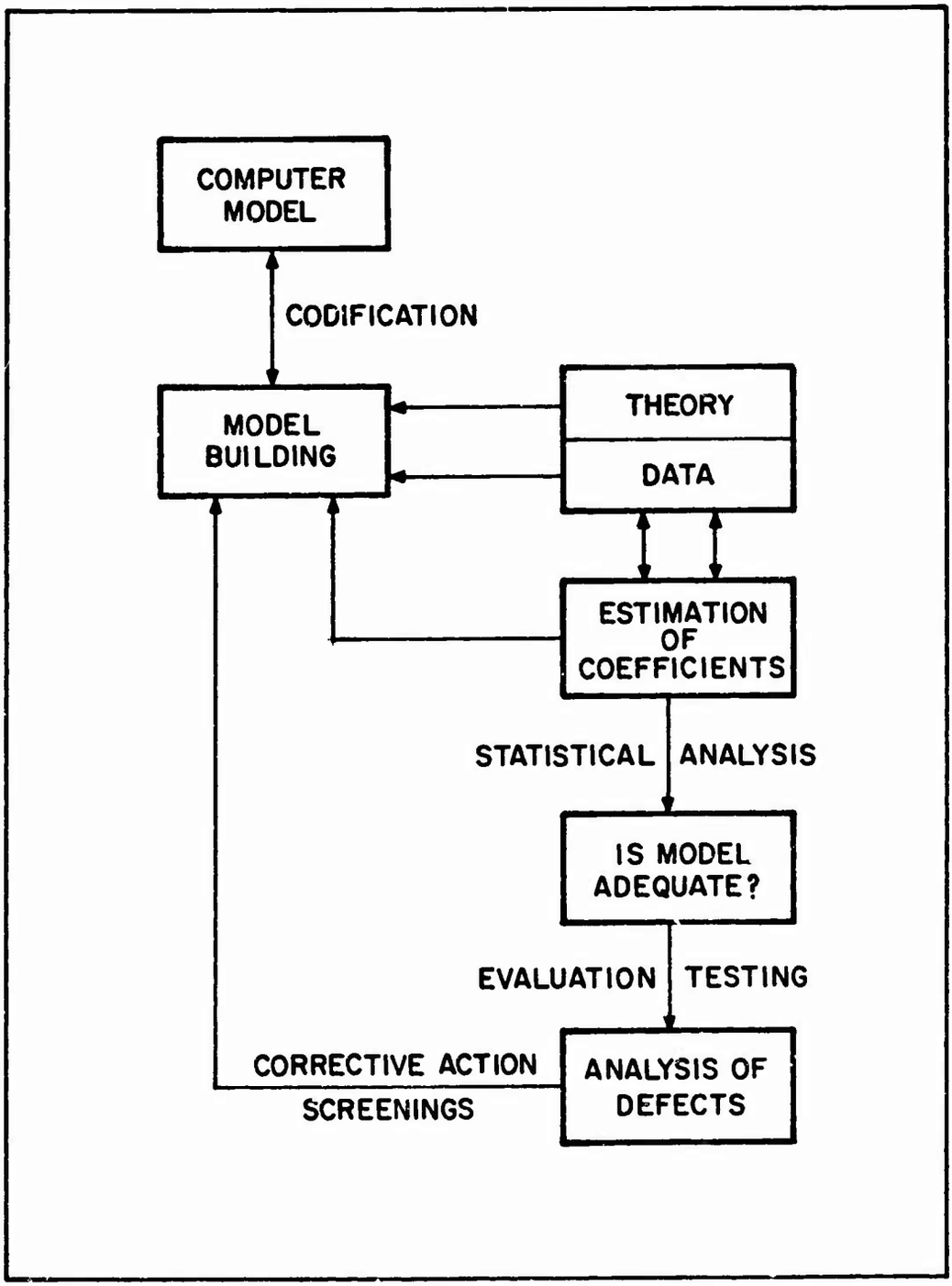


Fig. 1 ADAPTIVE MODEL BUILDING

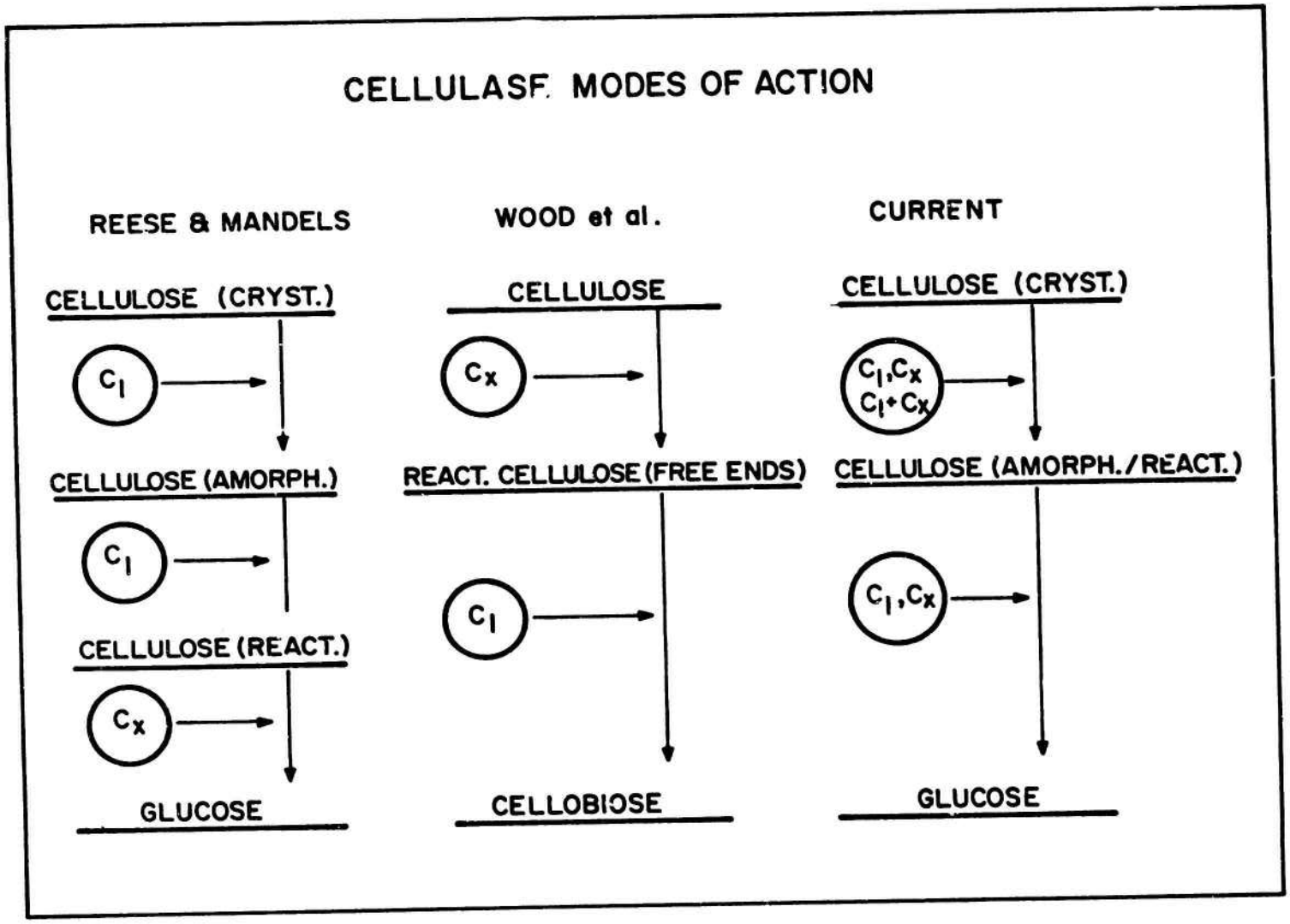

CELLULASE MODES OF ACTION
REESE & MANDELS
CELLULOSE (CRYST.)
C1
CELLULOSE (AMORPH.)
C1
CELLULOSE (REACT.)
Cx
GLUCOSE
WOOD et al.
CELLULOSE
Cx
REACT. CELLULOSE (FREE ENDS)
C1
CELLOBIOSE
CURRENT
CELLULOSE (CRYST.)
C1,Cx
C1+Cx
CELLULOSE (AMORPH./REACT.)
C1,Cx
GLUCOSE

Fig. 2

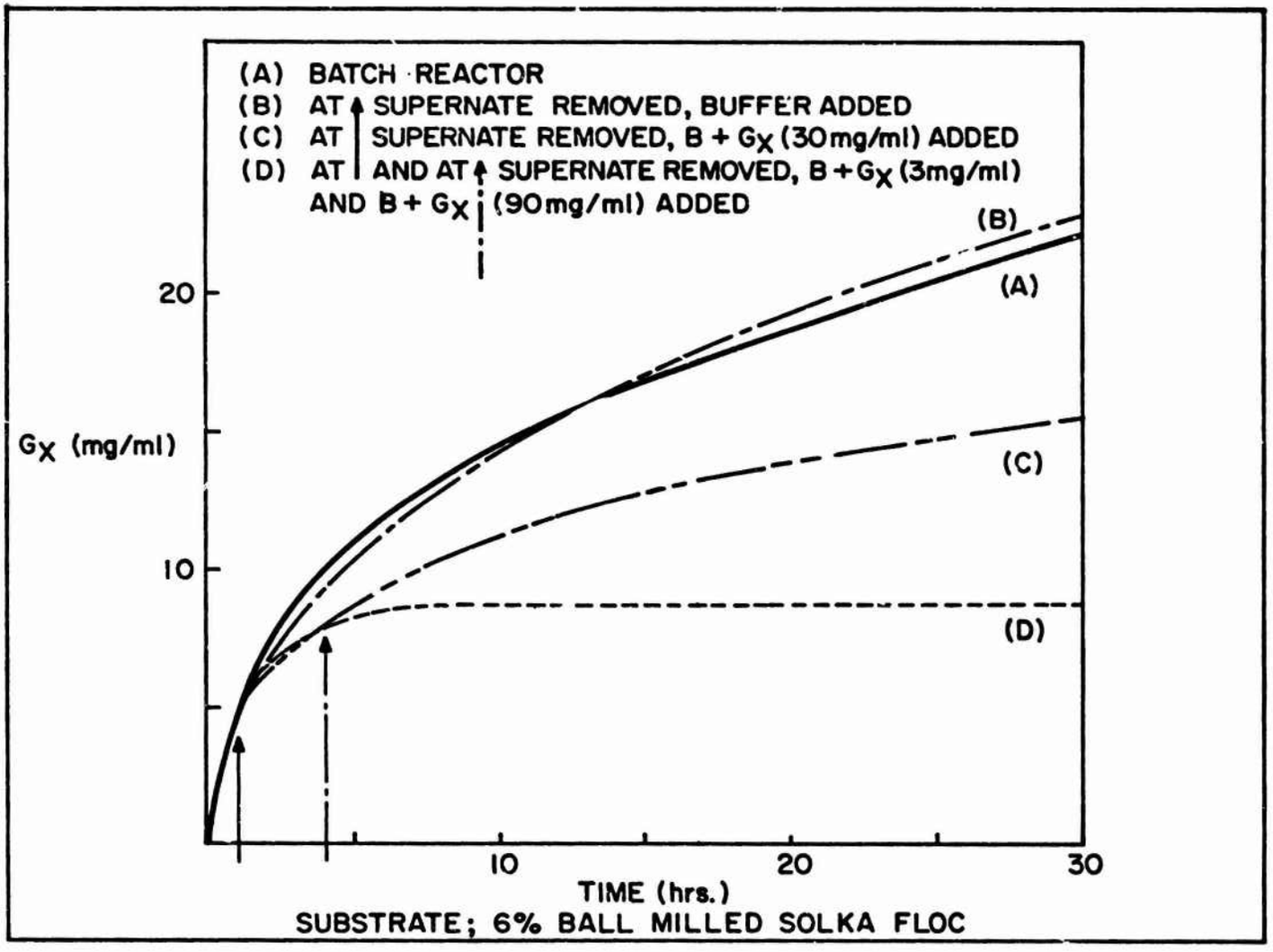

(A) BATCH REACTOR
(B) AT ↑ SUPERNATE REMOVED, BUFFER ADDED
(C) AT | SUPERNATE REMOVED, B + Gx (30mg/ml) ADDED
(D) AT | AND AT ↑ SUPERNATE REMOVED, B + Gx (3mg/ml)
AND B + Gx | (90mg/ml) ADDED
(B)
(A)
(C)
(D)
20
10
Gx (mg/ml)
10
20
30
TIME (hrs.)
SUBSTRATE; 6% BALL MILLED SOLKA FLOC

Fig. 3

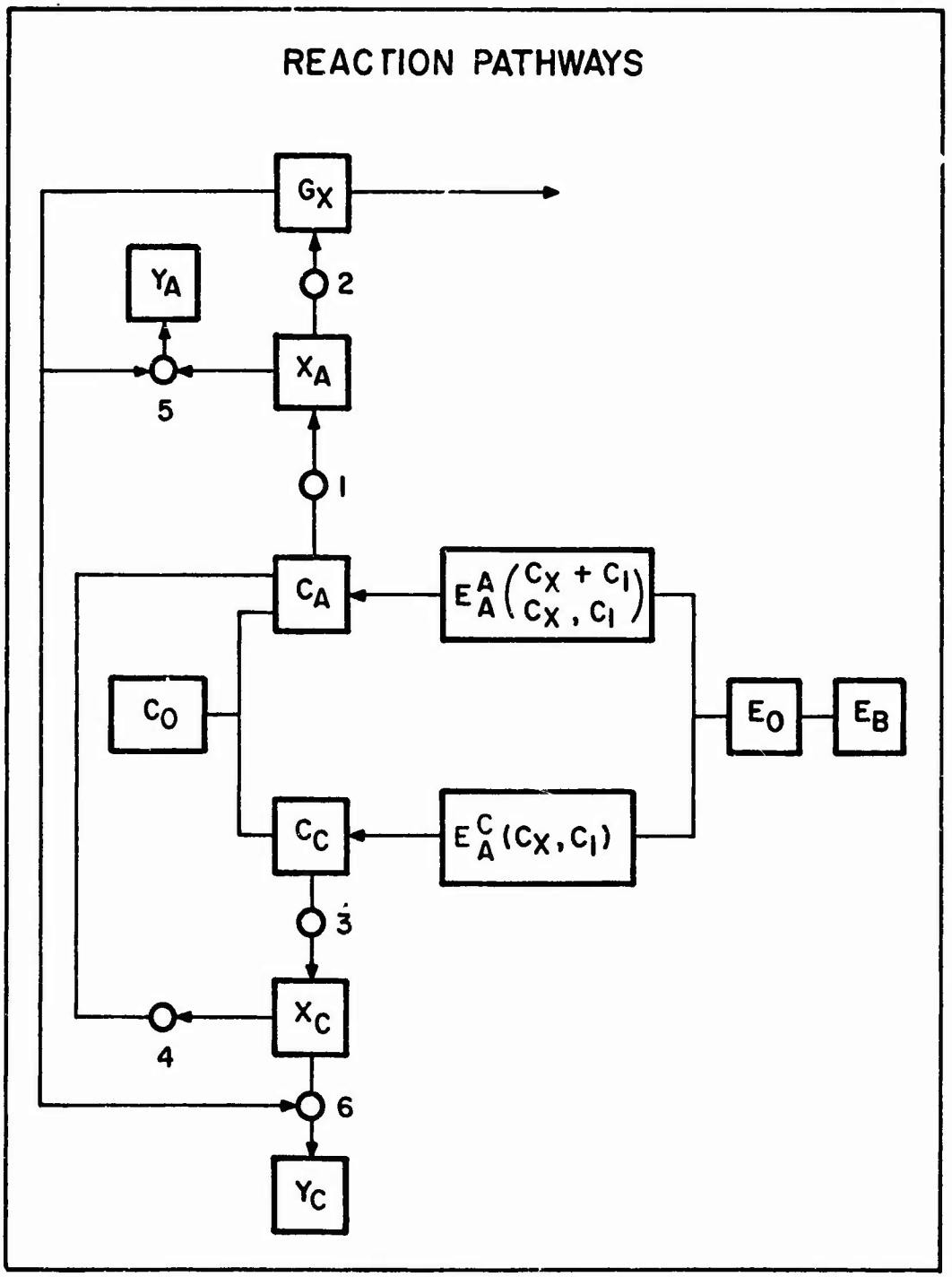

REACTION PATHWAYS
G_X
Y_A
2
X_A
5
1
C_A
$E_A^A\left(\begin{matrix}C_X + C_I\\ C_X, C_I\end{matrix}\right)$
C_O
E_O
E_B
C_C
$E_A^C(C_X, C_I)$
3
X_C
4
6
Y_C

Fig. 4

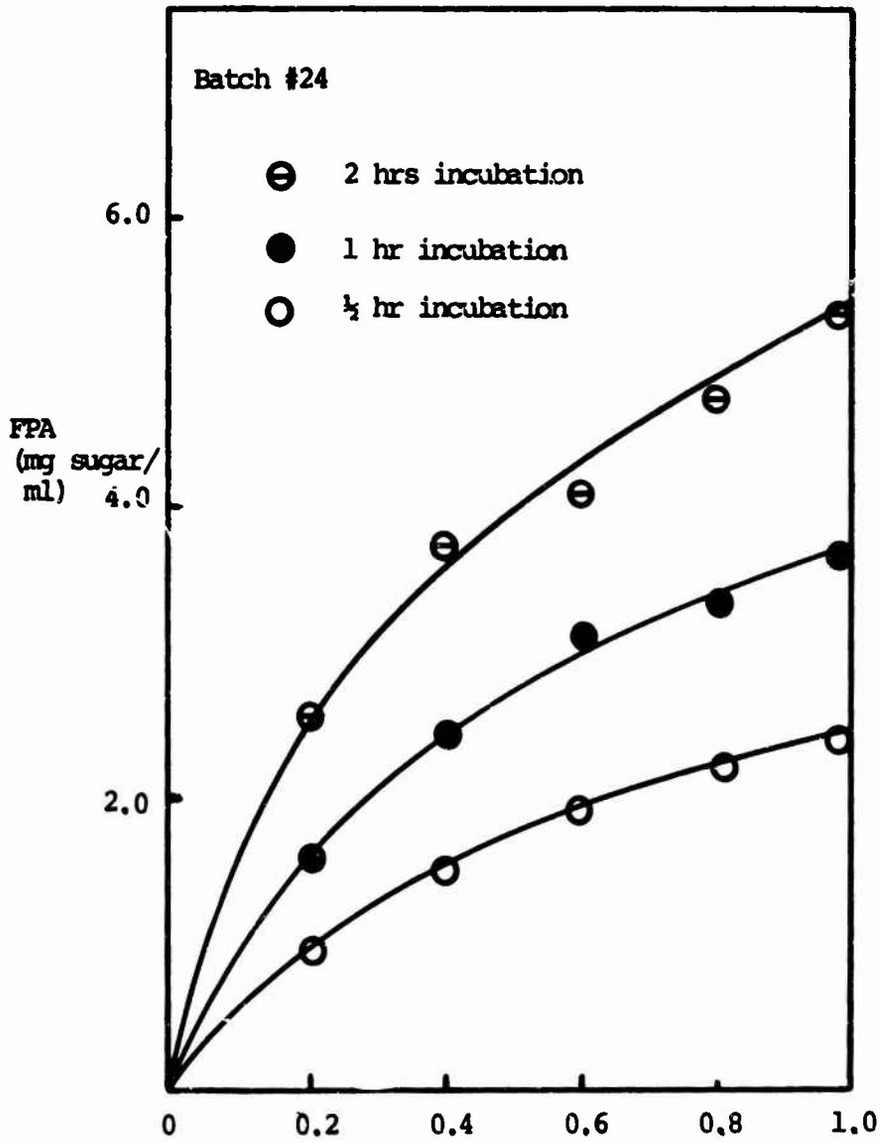


Fig 5 FPA vs. dilution (d)

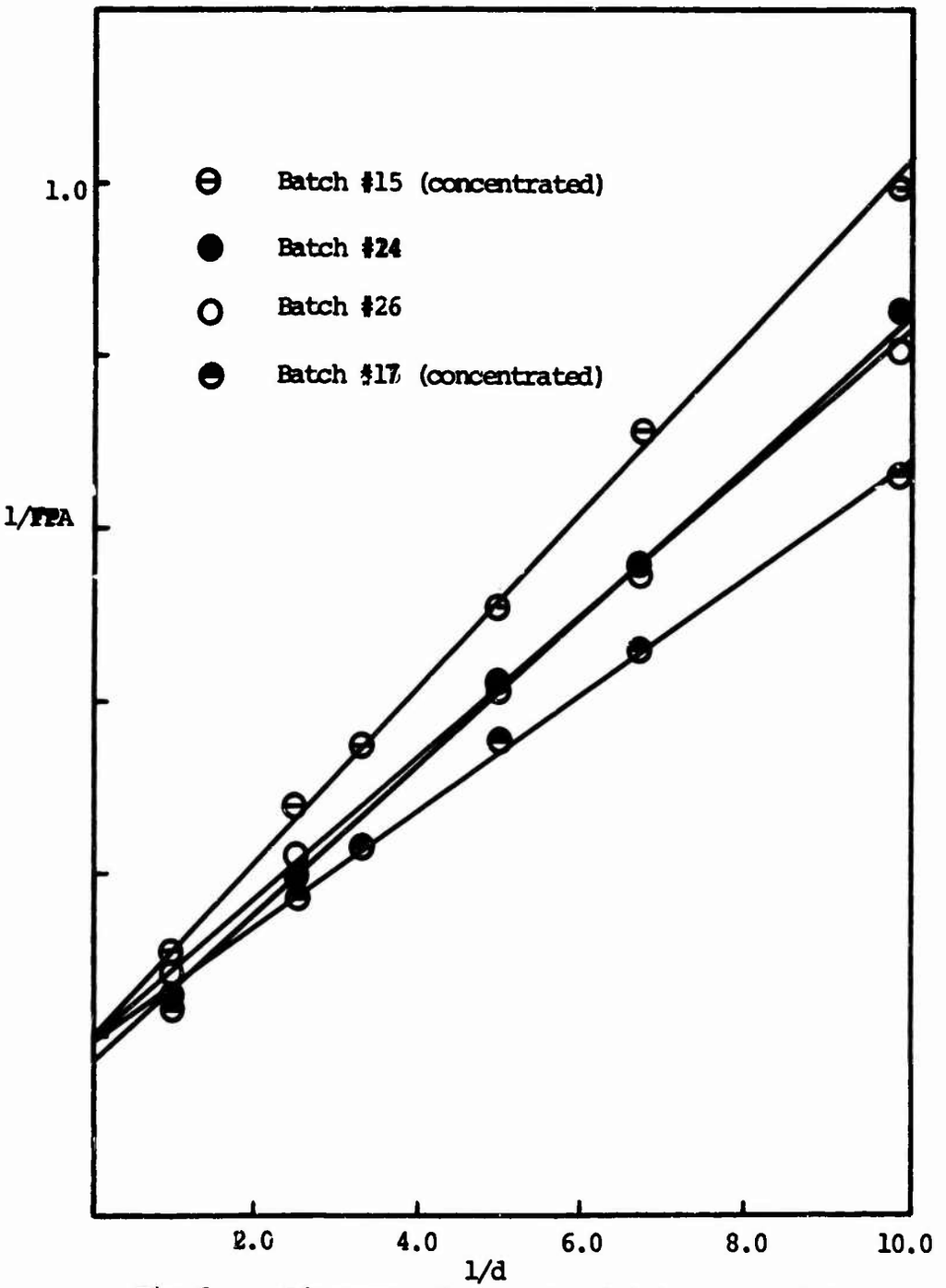


Fig 6a Lineweaver-Burke type plot for FPA and d

Figures 6a and 6b were prepared from the data obtained from Dr. M. Mandels

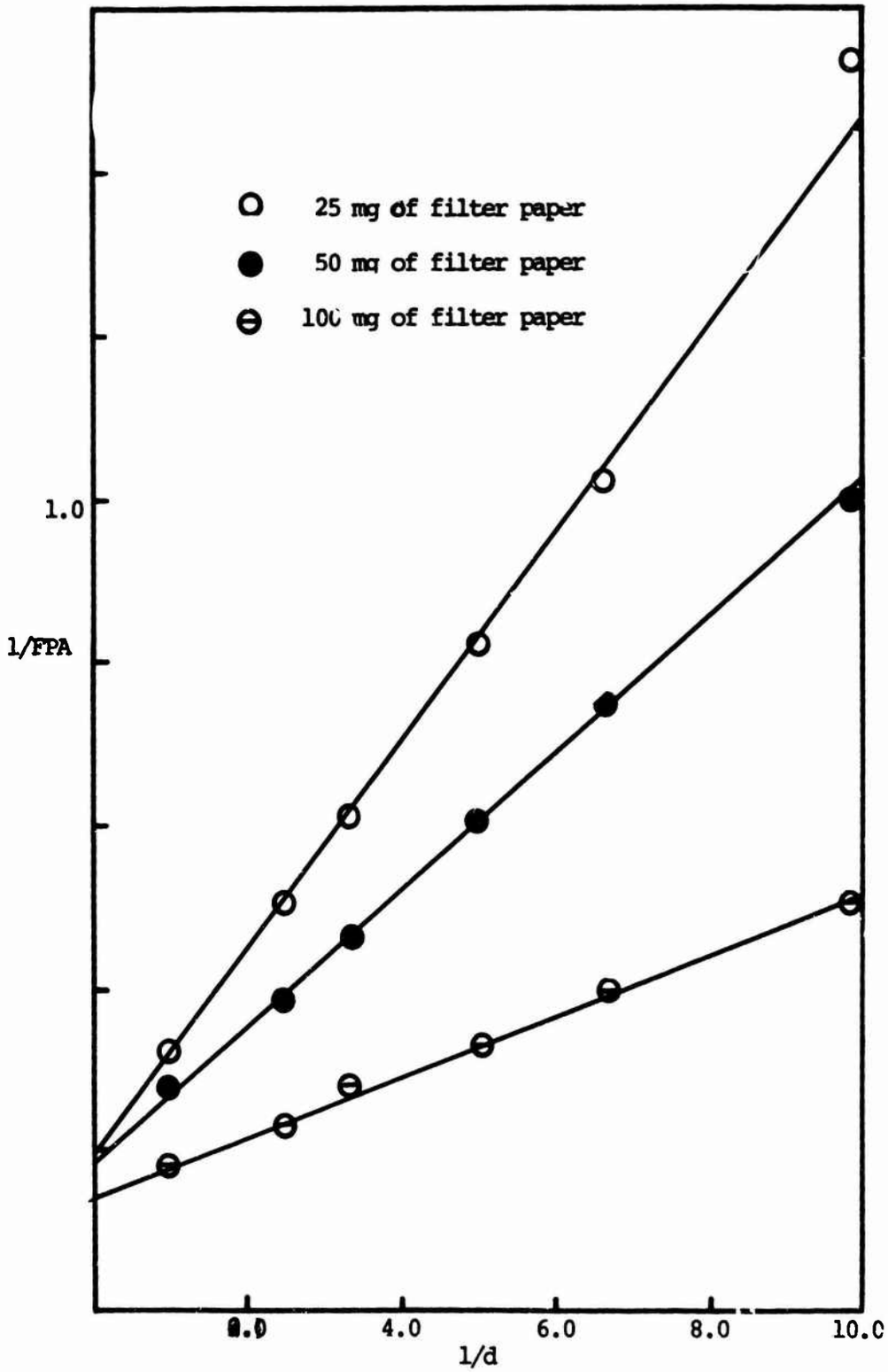


Fig 6b Lineweaver-Burke type plot for FPA and d

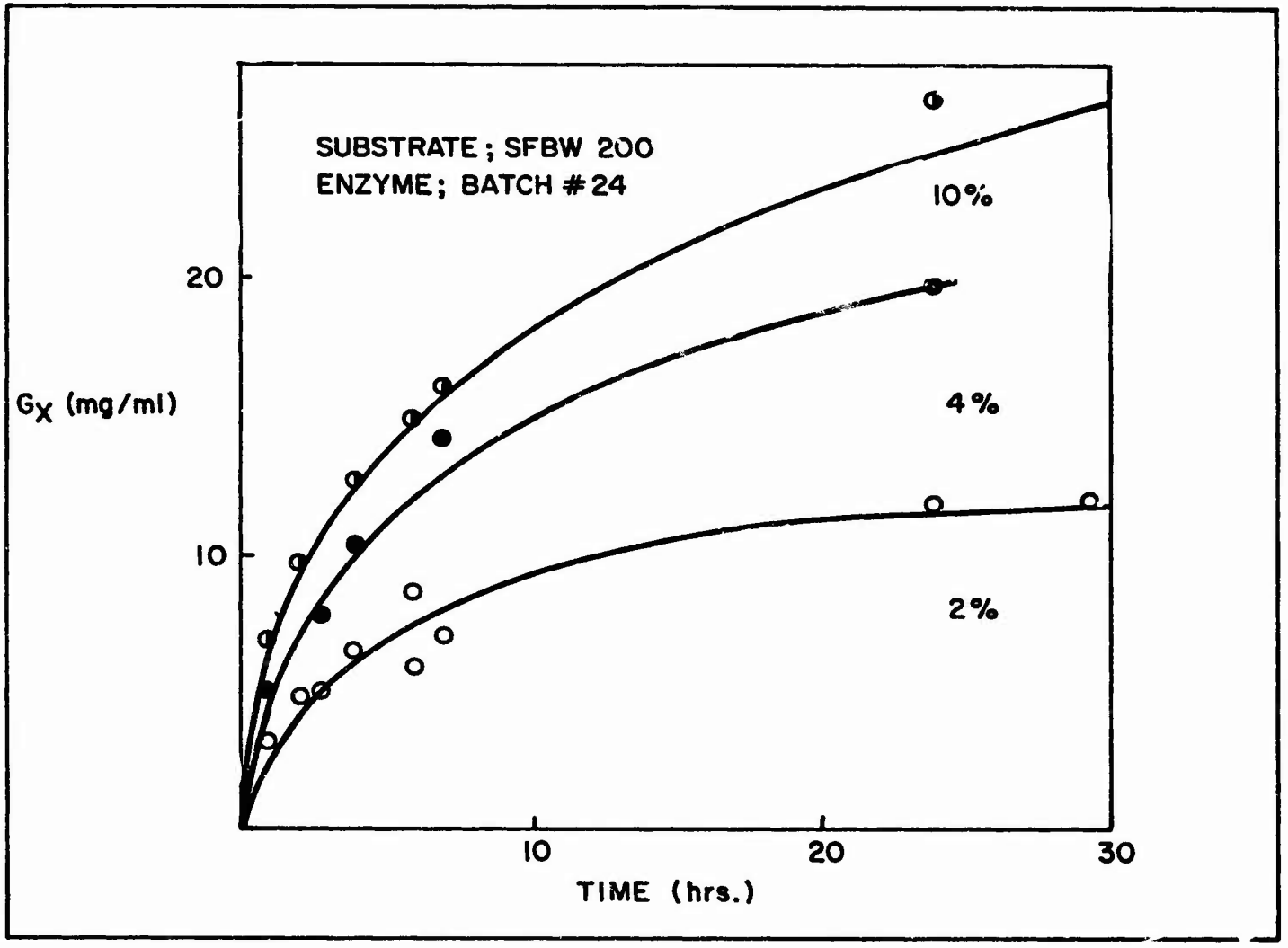

SUBSTRATE ; SFBW 200
ENZYME; BATCH #24
10%
4%
2%
20
10
G_X (mg/ml)
10
20
30
TIME (hrs.)

Fig. 7

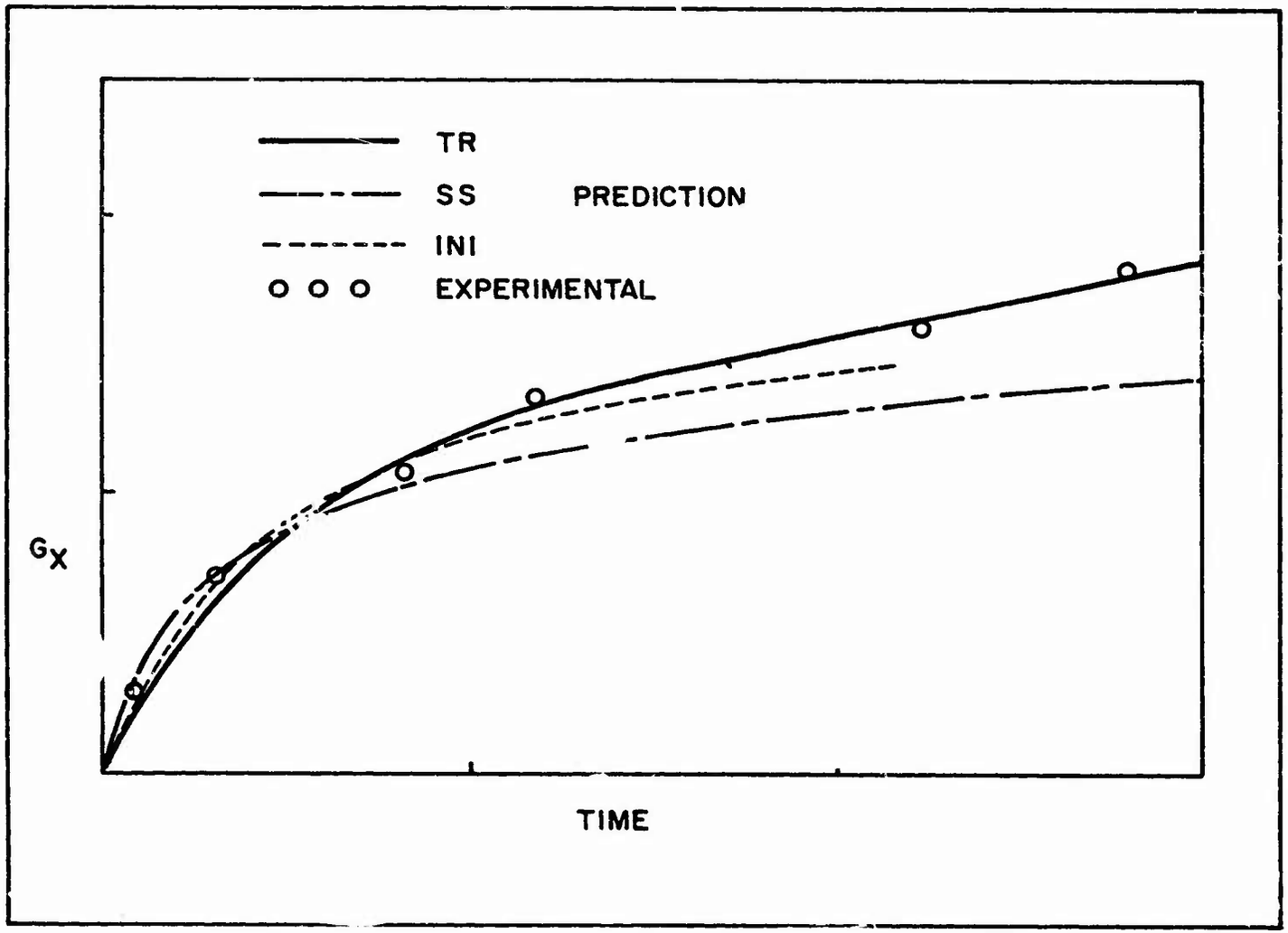


Fig. 8

TABLE A

Effect of the amount of filter paper on B_1 and B_2 in Eq. A

Filter Paper (mg)	B_1	B_2
25	5.0	0.89
50	5.3	0.63
100	7.1	0.44

Cellulase Batch #24 (QM 9414, SW 40)
Pr = 1.4 mg/ml at full strength
FPA = 3.68 mg sugar/ml at full strength

TABLE B

B_1 and B_2 for different culture batches of cellulase

Batch #	FPA (mg sugar/ml)**	Pr (mg/ml)	B_1	B_2
24	3.68	1.4	5.3	0.63
26	3.67	1.5	4.9	0.61
15*	3.39	1.5	4.9	0.76
17* (NP)	4.60	2.72	5.0	0.92

* concentrated
** at full strength
QM 9414 = second generation of Tv fungus mutant QM 9123
SW 40 = pure cellulose, NP = news paper used as growth medium

The FPA and Pr data were obtained from Dr. M. Mandels of this lab.

COMPUTED ENERGY DISTRIBUTIONS OF DOUBLE-SCATTERED PHOTONS OBTAINED FOR PURPOSES OF MINE DETECTOR DESIGN ANALYSIS

Fredrick L. Roder and Douglas G. Conley
Mine Detection Division, Countermine/Counter Intrusion Department
U. S. Army Mobility Equipment Research and Development Center
Fort Belvoir, Virginia

ABSTRACT. The backscatter of low-energy x and gamma photons has received considerable attention over the past several years as a possible technique for the detection of shallowly buried nonmetallic land mines. In the course of this work it was established experimentally that the sensitivity of a backscatter mine detector increased as the fraction of singly scattered photons in the total backscatter radiation field decreased. Although not experimentally verifiable, it was presumed, based on the literature, that (excluding single scatter) double scatter constituted the major portion of the backscatter field. In attempting to verify this presumption, use was made of a HP 2100 computer. By mathematically modelling the case of a monoenergetic collimated gamma-ray source and collimated detector, the energy distribution of doubly scattered photons was computed for several incident energies and scattering angles. By comparison of these results with experimentally obtained data, it was concluded that double scatter did not represent an appreciable component of the backscatter radiation field.

1. INTRODUCTION. The spectrum of x and gamma radiation backscattered from a scattering medium is a function of the chemical composition of that medium. Photon backscatter has therefore received considerable attention over the past several years (in both applied research and advanced development) as a possible technique for detecting shallowly buried non-metallic mines. The present paper concerns one aspect of this investigation.

Figure 1 depicts the experimental set up utilized in the early phases of the program. The source and detector were both highly collimated and coplanar. The collimator axes were positioned to intersect within or below the mine target. The angle included at the intersection of the collimator axes we shall call Δ.

Figure 2 shows typical pulse-height spectra obtained in this manner, utilizing a ^{137}Cs (662-keV) gamma source and setting $\Delta \simeq 140^{o}$. The simulated mine (solid curve) was in this case a 1 lb. block of DNB, a substance chemically similar to TNT, buried at a depth of 1 in. The peak which may be discerned in these spectra at ~200 keV is due to the presence of back-scattered photons which have scattered only once in the medium. The energy of such photons is uniquely determined by the energy of the incident gamma and the included angle Δ. However, no such simple relationship exists to predict the energy distribution of photons scattered two or more times in the medium.

On the basis of experiments of this type, it was determined that when a target was encountered, the change in the backscatter spectrum increased as Δ (the angle included between the collimator axes) decreased. This is equivalent to saying that sensitivity to the presence of a target increased as the ratio of single-scattered photons to the total photon backscatter decreased. It was therefore considered desirable to learn something about the origin of these other-than-single-scattered photons.

A logical tact for such an investigation to take would be to presume that a sizable portion of these photons had undergone two scattering events within the medium. This presumption was reinforced by the reported results of E. Hayward and J. Hubbell[1] of NBS, who had performed similar experiments and had concluded that the large number of photons to be found at energies below the single-scatter peak were primarily the result of double scatter.

The approach which we chose to verify this presumption was computational and was executed with the aid of a Hewlett-Packard 2100A computer.

2. CALCULATION. Figure 3 illustrates the sequence of events modelled by our program. Two parameters were initially specified: The energy Eo of the photons emitted by the source, and θ, the Compton single-scatter angle, defined at the intersection of the source and detector collimator axes. We have assumed the collimation of the source and detector to be sufficiently tight as to neglect angular dispersion of the collimator acceptance cones in the vicinity of their crossover point. For the given input conditions, a number of photon histories were compiled. This number was twice the number of first-scatter angles θ_1 considered, since, as can be seen from the figure, for each value of θ_1, there are two alternate paths to acceptable second-scatter points. Values of θ_1 were selected by taking equal increments of $\cos\theta_1$ along the interval +1 to -1. The point of the first scatter remained arbitrary, as the factors which would require its specification, i.e., mass attenuation and the solid angle subtended by the detector collimator cone, were neglected in this calculation. Similarly, while the point of second scatter remained arbitrary, the second-scatter angle for each of the two photon paths shown was obtained from the condition that $\theta_1+\theta_2=\theta$. Using these paired values of θ_1 and θ_2, together with the incident photon energy Eo, the spectral distribution for double scattered photons was then calculated.

The cross section in barns for a photon of energy E scattering through an angle θ is given by the Klein-Nishina formula:

[1] Evans Hayward and John H. Hubbell, "An Experiment on Gamma-Ray Backscattering," NBS Report No. 2264 (1953); J. Appl. Phys. 25, 506 (1954).

$$\frac{d\sigma_c}{d\Omega} = r_o^2 \left[\frac{1}{1+\alpha(1-\cos\theta)}\right]^3 \left(\frac{1+\cos^2\theta}{2}\right) \left[\frac{\alpha^2(1-\cos\theta)^2}{1+(1+\cos^2\theta)\{1+\alpha(1-\cos\theta)\}}\right], \quad (1)$$

where $\alpha=E/m_oc^2$, r_o is the classical radius of the electron, and m_o is the mass of the electron. The energy of the scattered photon E′ is related to its incident energy E and the angle of scatter by the Compton formula:

$$E' = \frac{E}{1+\alpha(1-\cos\theta)} \quad (2)$$

Utilizing these expressions, the spectral distribution of double-scattered photons was computed by a program conforming to the block diagram shown in Fig. 4.

a. Initially, the single-scatter angle θ, incident photon energy E_o, and $\cos\theta_1$ step size were inputted.

b. For each $\cos\theta_1$ value, two values for θ_2 were determined.

c. The Compton and Klein-Nishina formulas were employed to determine the cross section σ_{c1} for scattering through an angle θ_1 and to determine the energy E_1 of the scattered photon. E_1 and θ_2 were then substituted in the same formulas to obtain σ_{c2} and E_2.

d. The energy bin corresponding to E_2 was then incremented by the product of σ_{c1} and σ_{c2}. Two hundred fifty such energy bins were available, each 2.5-keV wide, and spanning the interval from 0 to 625 keV.

e. Blocks 2, 3, and 4 were repeated for all values of θ_1. The accumulated value of $\sigma_{c1} \times \sigma_{c2}$ in each energy bin was then a measure of the relative probability of a photon of energy E_o scattering twice in the medium and arriving at the detector with an energy falling within that bin.

f. The final step was to output in both graphical and tabular formats the accumulated values for $\sigma_{c1} \times \sigma_{c2}$ as a function of the lower energy limit of each bin.

3. RESULTS. The spectra of double-scattered photons were computed by means of the above-discussed program for the set of input parameters shown in Fig. 5. 1.25 MeV is the average energy of the 1.17 and 1.33 MeV gammas produced in equal numbers by ^{60}Co; 0.662 MeV is the energy of a ^{137}Cs gamma; and 0.122 MeV is the energy of the 87% yield gamma from ^{57}Co. The double-scatter spectra for ^{60}Co for $\theta=60^o$ and 30^o were not obtained, as the energy maximum for these spectra exceeded the 625 keV limit incorporated into the program.

Figure 6 shows the spectrum obtained for E_0=1.25 MeV and θ=90°. Most notable in this spectrum is the unexpected presence of sharp maxima at the maximum and minimum possible energies, energies which correspond to angle pairs wherein $\theta_1 = \theta_2$. The presence of these maxima is not due to exceptional values of $\sigma_{C1} \times \sigma_{C2}$ for $\theta_1 = \theta_2$, but is rather due to the slow change of E with $\cos\theta_1$ about these two points. As an illustration, in Fig. 7 we have plotted E as a function of $\cos\theta_1$ for $1 \leq \cos\theta_1 \leq 0$ and $0 \leq \cos\theta_2 \leq 1$, again setting E_0=1.25 MeV and θ=90°. Note that E values fall within the narrow range from 512.5 to 513.78 keV (corresponding to the highest energy bin of the spectrum shown in Fig 6) for approximately 8% of all $\cos\theta_1$ values, whereas only approximately 1.25% of E values fall within a 2.5-keV-wide bin for lower E values. Figure 8 depicts the double-scatter spectra for E_0=1.25 MeV and θ=90°, 120°, and 150°. Note that as θ increases the total energy bandwidth of the spectrum decreases. In the limit of θ=180° all double-scattered photons would be found at a single energy, in this case 212 keV.

Figures 9 and 10 depict, respectively the double-scatter spectra obtained for E_0= 0.662 and 0.122 MeV for θ values of 30°, 60°, 90°, 120°, and 150°. From these it may be observed that as E_0 decreases, so does the energy bandwidth and median energy of the double-scatter spectrum.

4. CONCLUSIONS. The one salient result of the present effort is the predicted existence of a well-defined maximum in the double-scatter spectrum at an energy well above the single-scatter energy. As such, it should be observable in a portion of the backscatter energy spectrum which would otherwise be clean, and consequently it should be readily observable even with conventional scintillation detectors.[2]

Since such a maximum is absent in experimentally obtained backscatter spectra, such as those shown in Fig. 2, we must conclude that double scatter does not contribute appreciably to these spectra. The portion of these spectra below the single-scatter peak we therefore conclude must arise from multiple-scattered photons.

In practical terms, this conclusion frees the designer of the mine detector searchhead from the requirement that source and detector collimators be coplanar, since while all double scatters occur within a plane, multiple scatters do not.

5. ACKNOWLEDGMENTS. The authors wish to express their appreciation to Charles Eisenhauer and Louis Spencer of the National Bureau of Standards for their invaluable guidance throughout the course of this effort, to Louis Mittelman for his aid in the programming, and to Robert L. Brooke and Karl H. Steinbach, under whose supervision this work was accomplished.

[2]The lower-energy maximum occurs in an energy region partly populated by multiple scattered photons and obscured by the low-energy (Compton) tail of the single-scatter peak. As such, it might realistically be expected to be obscured.

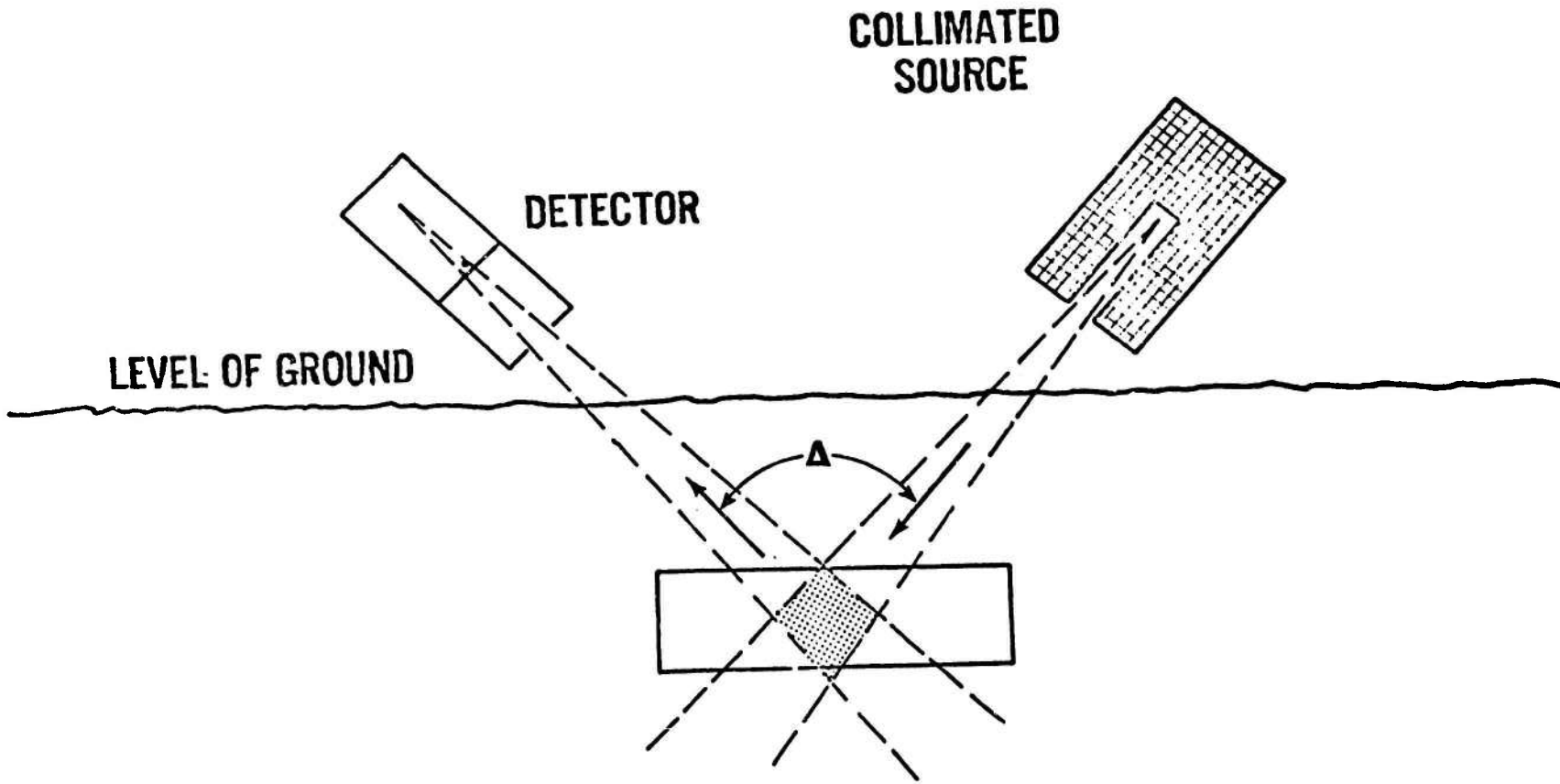


Figure 1

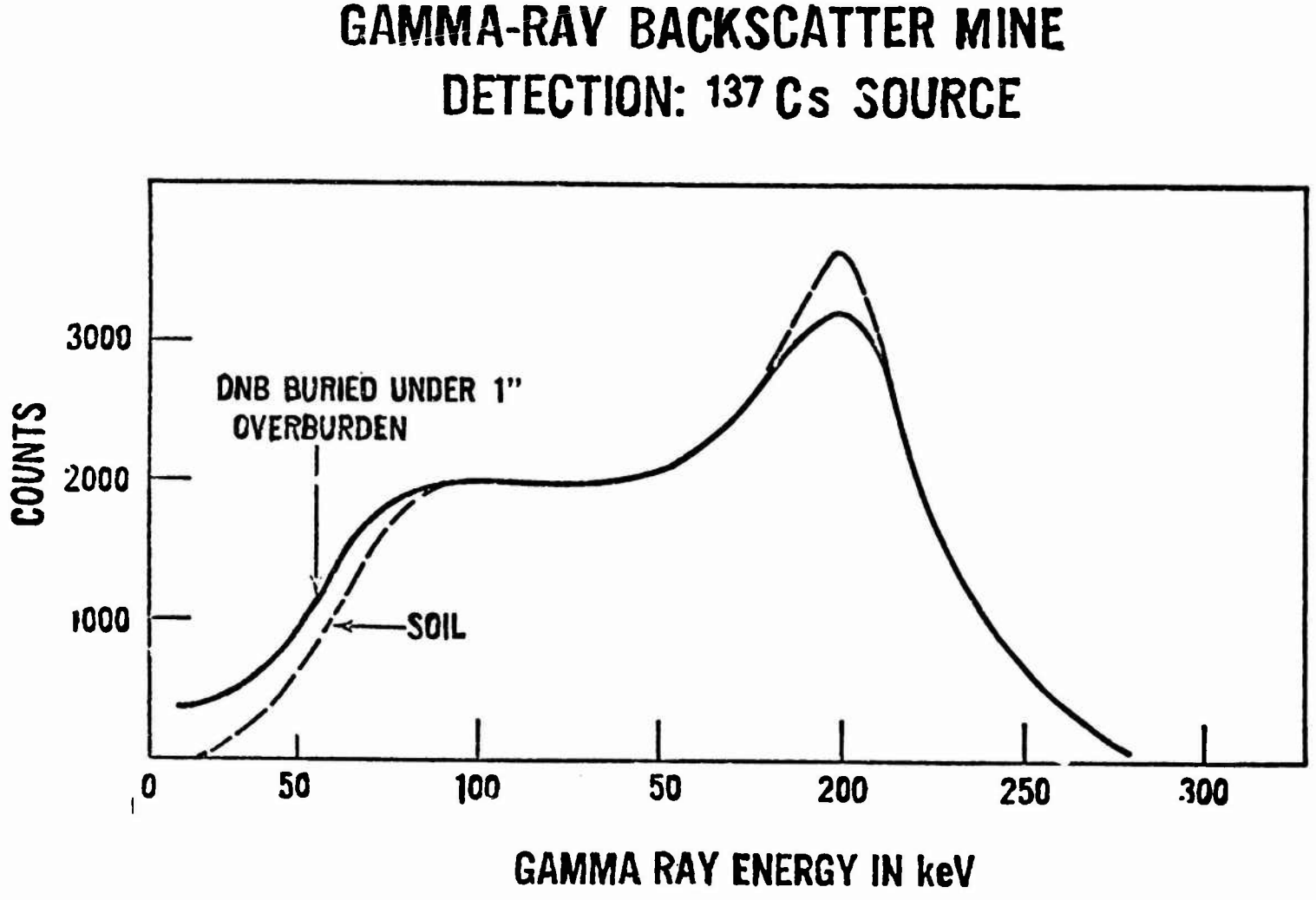

GAMMA-RAY BACKSCATTER MINE
DETECTION: 137 Cs SOURCE
COUNTS
3000
2000
1000
DNB BURIED UNDER 1"
OVERBURDEN
SOIL
0
50
100
50
200
250
300
GAMMA RAY ENERGY IN keV

Figure 2

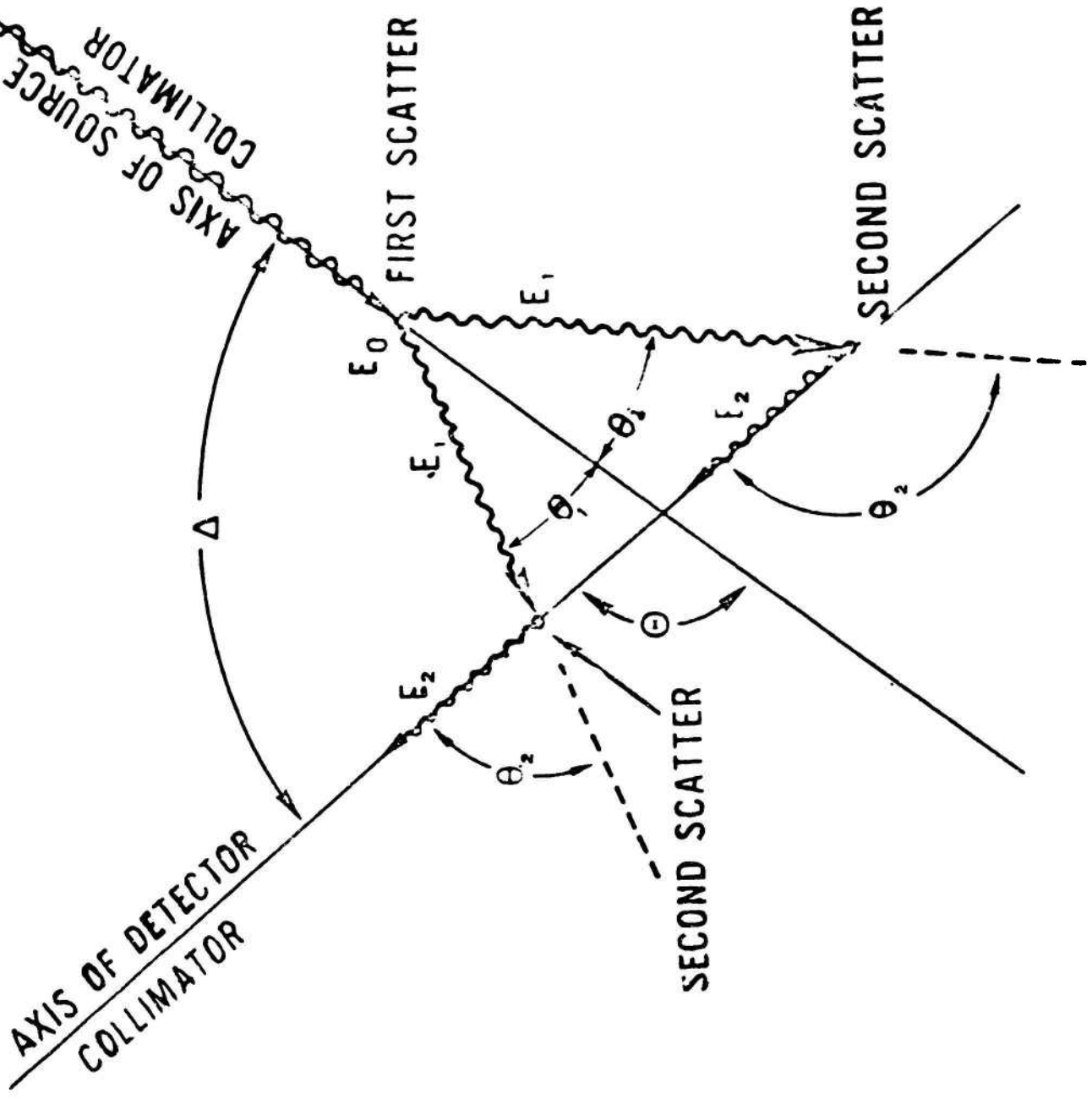

AXIS OF SOURCE COLLIMATOR
FIRST SCATTER
SECOND SCATTER
E_0
E_1
E_2
E_1'
E_2'
θ_1
θ_2
θ_1'
θ_2'
Θ
Δ
SECOND SCATTER
AXIS OF DETECTOR COLLIMATOR

Figure 3

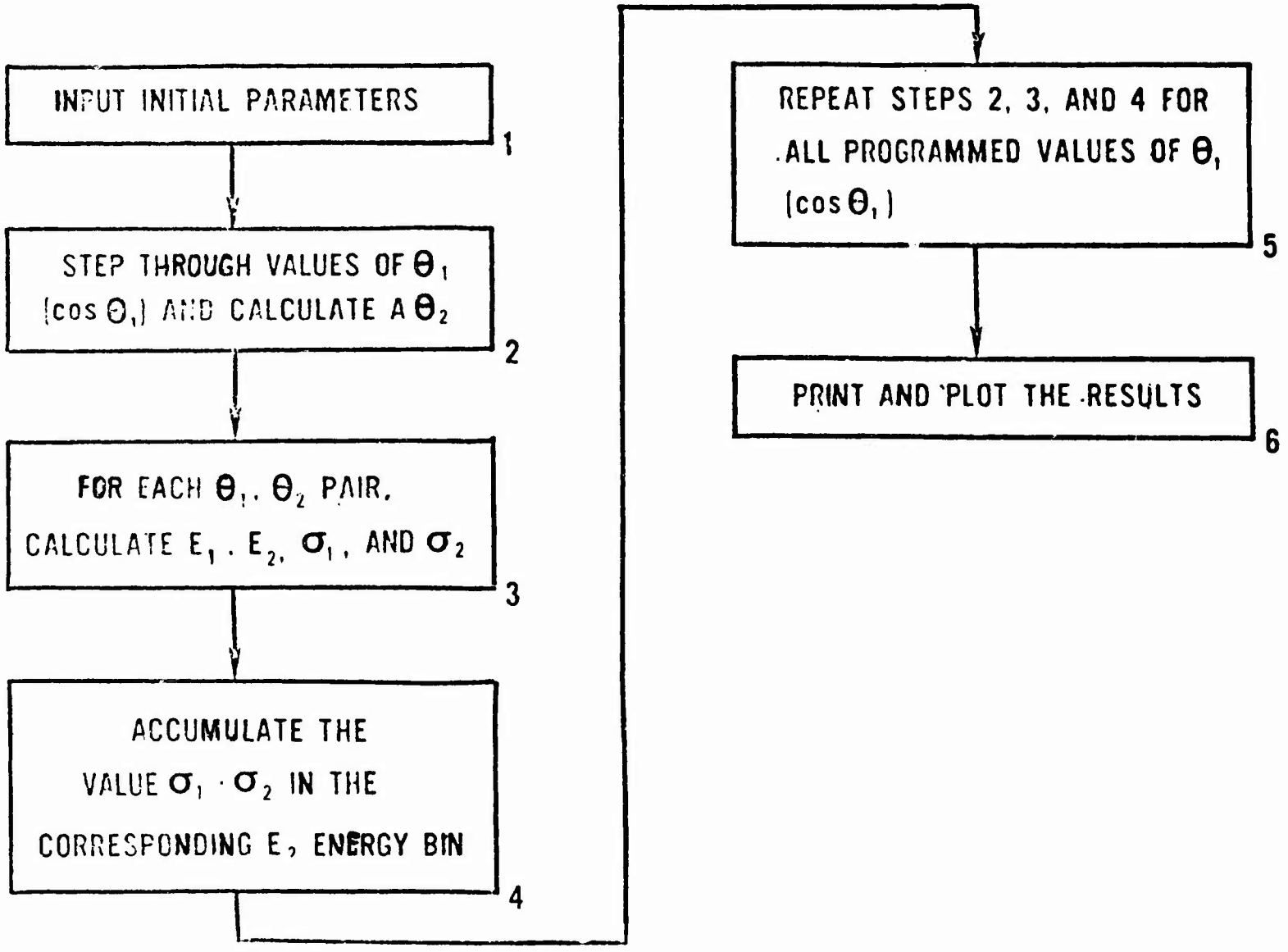

INPUT INITIAL PARAMETERS
1
STEP THROUGH VALUES OF θ_1 ($\cos\theta_1$) AND CALCULATE A θ_2
2
FOR EACH θ_1, θ_2 PAIR, CALCULATE E_1, E_2, σ_1, AND σ_2
3
ACCUMULATE THE VALUE $\sigma_1 \cdot \sigma_2$ IN THE CORRESPONDING E, ENERGY BIN
4
REPEAT STEPS 2, 3, AND 4 FOR ALL PROGRAMMED VALUES OF θ_1 ($\cos\theta_1$)
5
PRINT AND PLOT THE RESULTS
6

Figure 4

INPUT PARAMETERS

ISOTOPIC SOURCE	INCIDENT PHOTON ENERGY (MeV)	SINGLE SCATTER ANGLE θ (DEGREES)
^{60}Co	1.25	90, 120, 150
^{137}Cs	0.662	30, 60, 90, 120, 150
^{57}Co	0.122	30, 60, 90, 120, 150

Figure 5

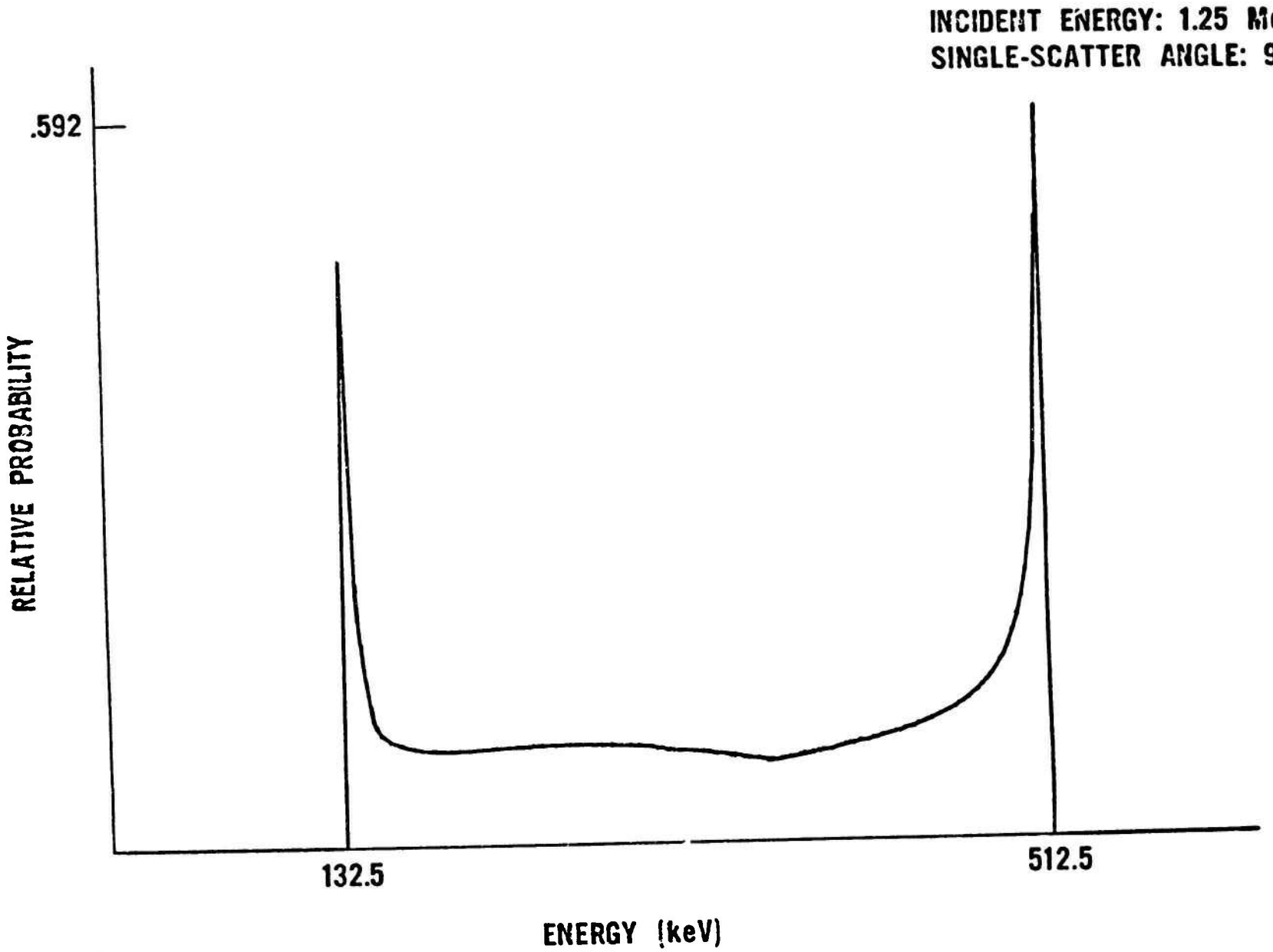

INCIDENT ENERGY: 1.25 MeV
SINGLE-SCATTER ANGLE: 90°
.592
RELATIVE PROBABILITY
132.5
512.5
ENERGY (keV)

Figure 6

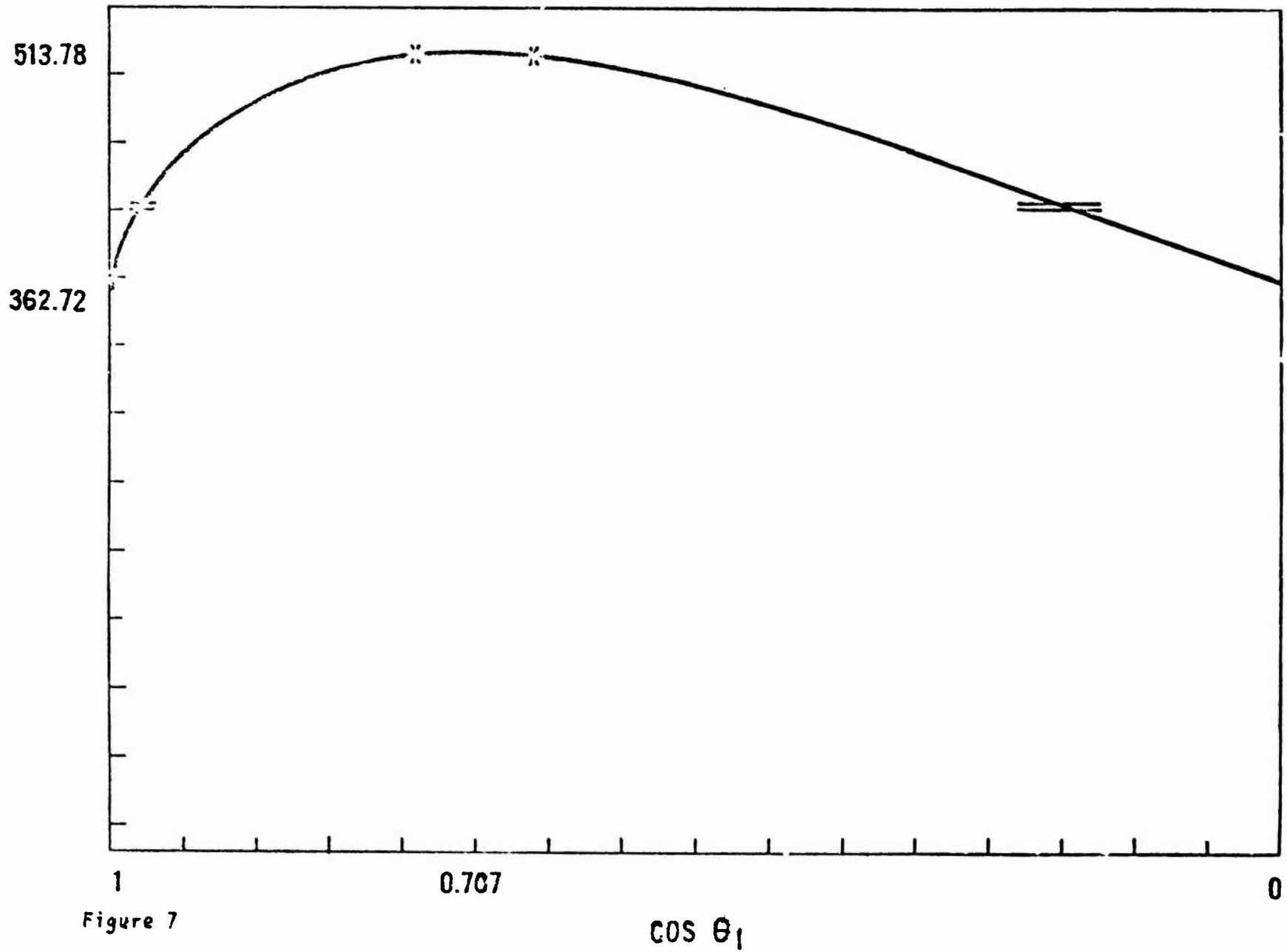
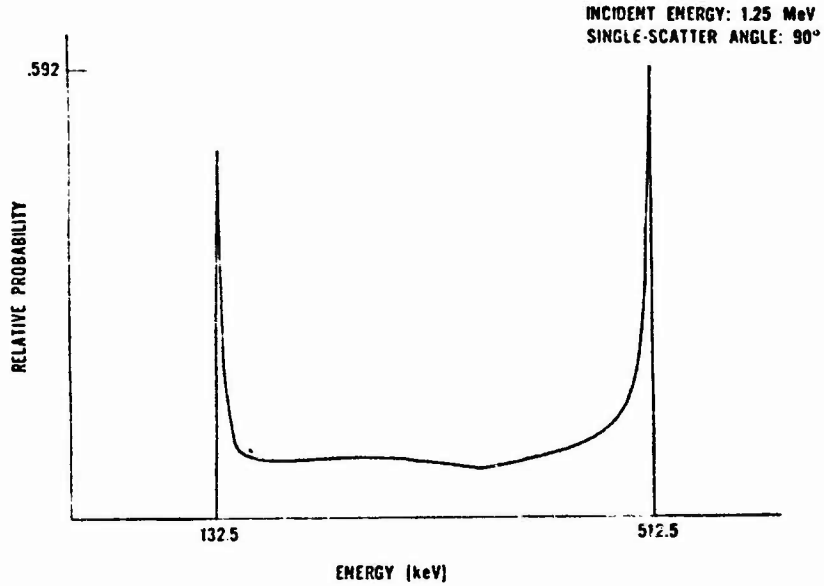
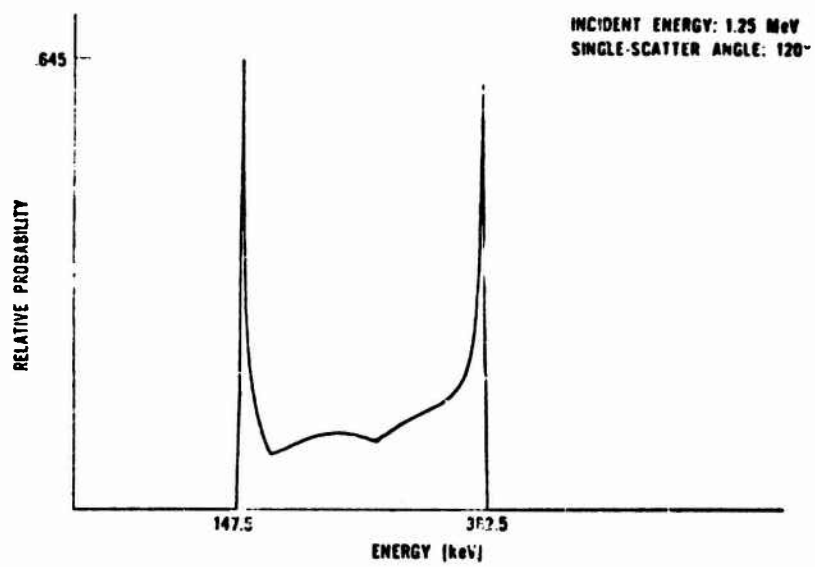


Figure 7


INCIDENT ENERGY: 1.25 MeV
SINGLE-SCATTER ANGLE: 90°
.592
RELATIVE PROBABILITY
132.5
512.5
ENERGY (keV)


INCIDENT ENERGY: 1.25 MeV
SINGLE-SCATTER ANGLE: 120°
.645
RELATIVE PROBABILITY
147.5
ENERGY (keV)

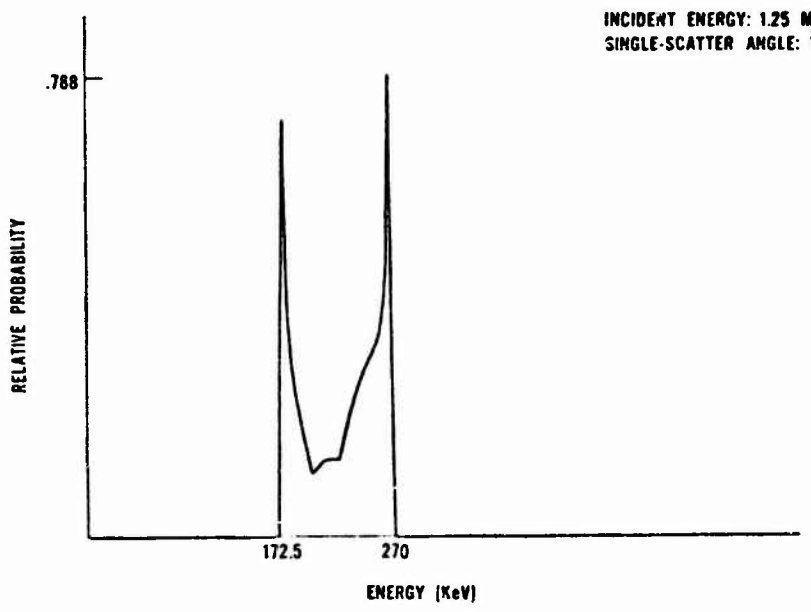

INCIDENT ENERGY: 1.25 MeV
SINGLE-SCATTER ANGLE: 150°
.788
RELATIVE PROBABILITY
172.5
270
ENERGY (KeV)

Figure 8

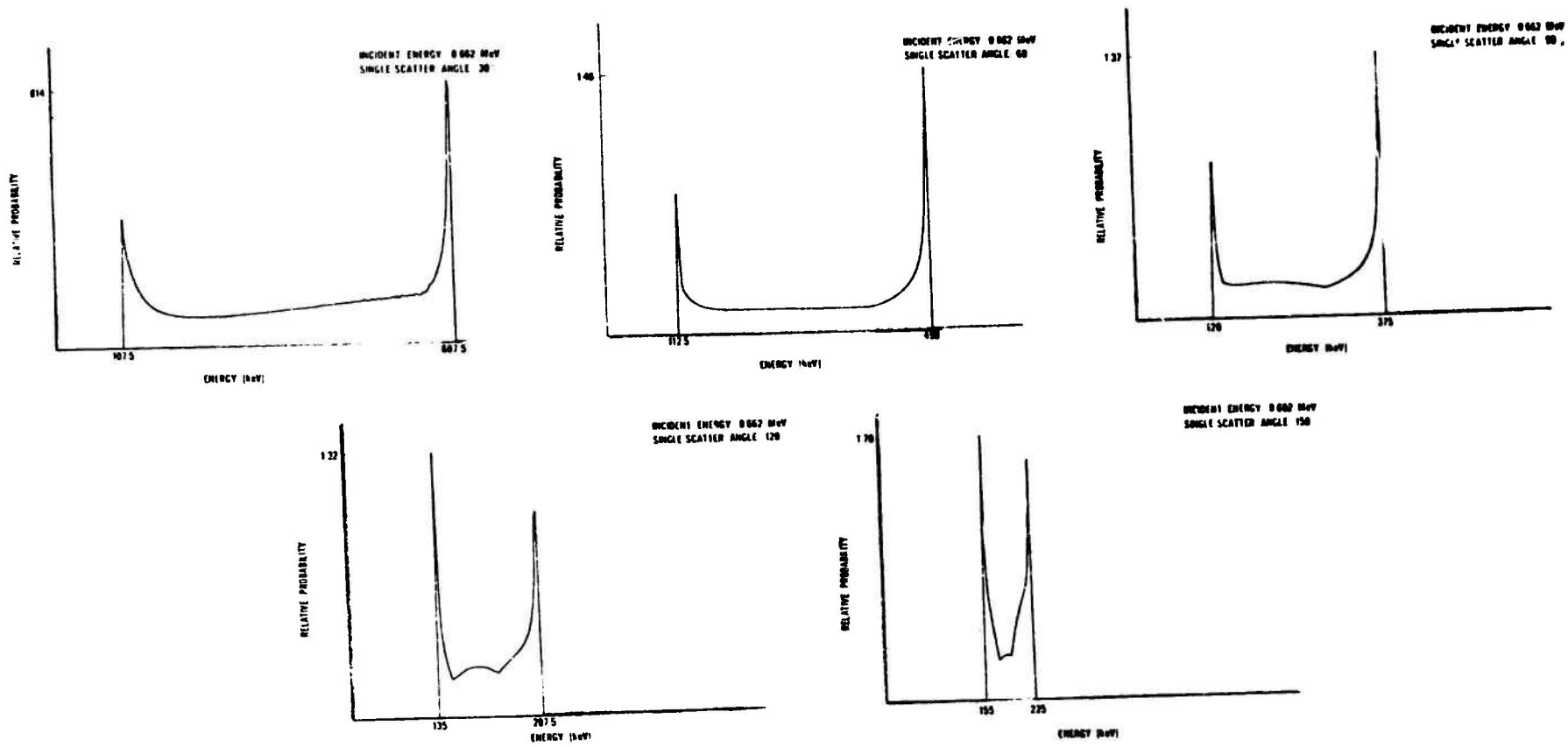

INCIDENT ENERGY 0.662 MeV
SINGLE SCATTER ANGLE 30
814
RELATIVE PROBABILITY
107.5
667.5
ENERGY (keV)
INCIDENT ENERGY 0.662 MeV
SINGLE SCATTER ANGLE 60
148
RELATIVE PROBABILITY
112.5
ENERGY (keV)
INCIDENT ENERGY 0.662 MeV
SINGLE SCATTER ANGLE 90
137
RELATIVE PROBABILITY
170
375
ENERGY (keV)
INCIDENT ENERGY 0.662 MeV
SINGLE SCATTER ANGLE 120
132
RELATIVE PROBABILITY
135
207.5
ENERGY (keV)
INCIDENT ENERGY 0.662 MeV
SINGLE SCATTER ANGLE 150
170
RELATIVE PROBABILITY
155
225
ENERGY (keV)

Figure 9

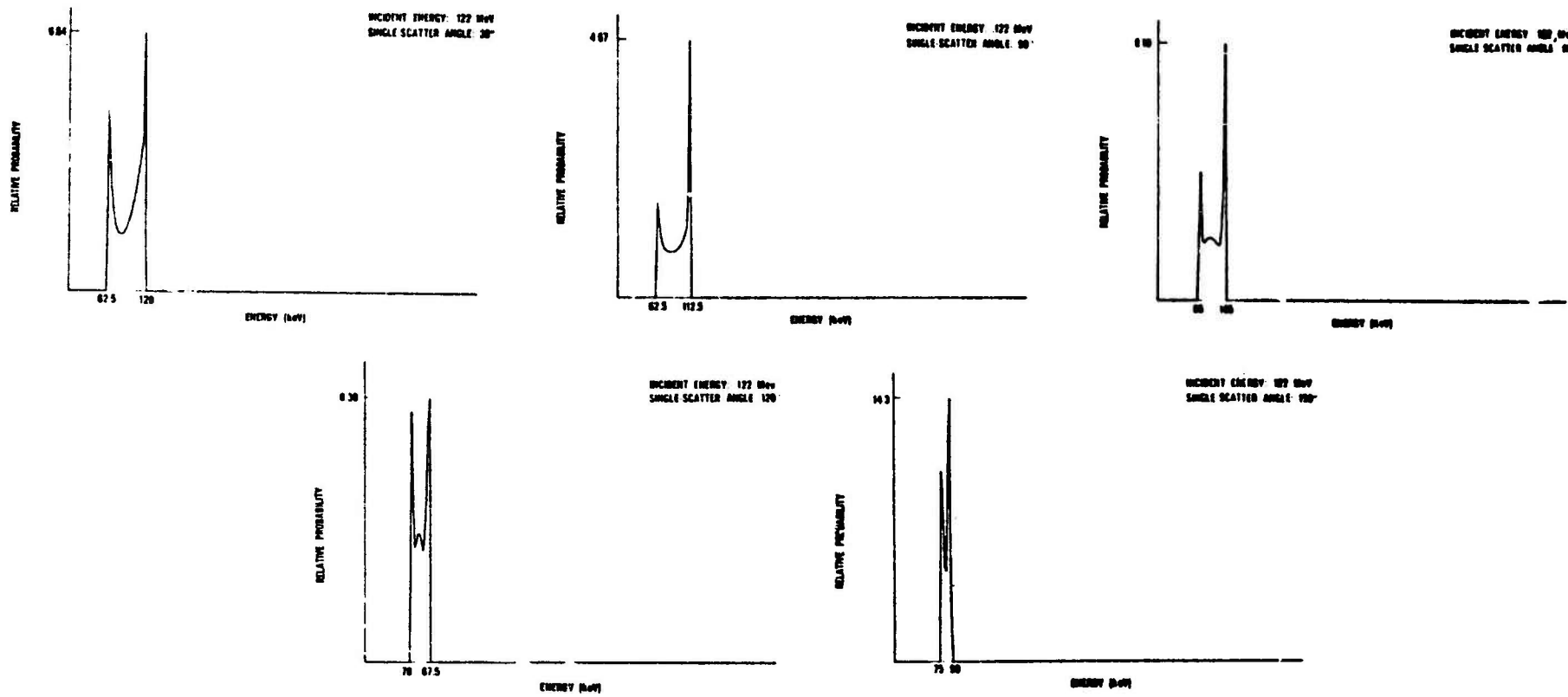

INCIDENT ENERGY: 122 kev
SINGLE SCATTER ANGLE: 30°
RELATIVE PROBABILITY
0.84
62.5
120
ENERGY (kev)
4.67
62.5
112.5
0.30
70
67.5
14.3
75
90

Figure 10

DOUBLE SCATTER ENERGY SPECTRUM

```
100   REM
101   REM
102   REM
103   REM      THIS PROGRAM CALCULATES THE ENERGY SPECTRUM
104   REM   RESULTING FROM DOUBLE SCATTER OF X AND GAMMA
105   REM   RADIATION IN A MEDIUM. THE FIRST SCATTER ANGLE IS
106   REM   STEPPED AS A FUNCTION OF THE COSINE. TWO SECOND
107   REM   SCATTER ANGLES ARE COMPUTED AND THE FINAL ENERGY
108   REM   PROBABILITY IS STORED IN THE APPROPRIATE BIN OF THE
109   REM   ENERGY ARRAY. THE OUTPUT IS A PLOT AND LIST OF THE
110   REM   SUM OF ALL PROBABILITIES IN A BIN VS. THE LOWEST ENERGY
111   REM   OF THE BIN.
112   REM
113   REM
114   REM          FRED RODER AND DOUG CONLEY
115   REM                19 NOVEMBER 1973
116   REM
117   REM
118   REM   SET THE NUMBER OF ENERGY BINS TO 250
120   DIM H[250]
122   FOR L=1 TO 250
124   LET H[L]=0
126   NEXT L
130   REM   INPUT CONSTANTS AND VARIABLES
132   LET P=3.14159
134   PRINT "WHAT IS THE SINGLE SCATTER ANGLE IN DEGREES"
136   INPUT X
138   LET X=X*P/180
140   LET Z=COS(X)
142   PRINT "WHAT IS THE INITIAL ENERGY IN MEV";
144   INPUT E
146   LET E=E/.511
148   PRINT "WHAT IS THE COSINE THETA ONE STEP SIZE";
150   INPUT T
152   REM
160   FOR I=1 TO 2
162   LIGHT(1,1)
164   IF I=2 LIGHT(3,1)
170   FOR T1=1 TO -1 STEP -T
172   IF I=2 GO TO 190
174   IF T1  0 LIGHT(2,1)
176   LET T2=Z*T1+SQR(Z12*T112-Z12-T112+1)
178   GOSUB 1004
180   NEXT T1
182   NEXT I
184   GOTO 1202
190   LET T2=Z*T1-SQR(Z12*T112-Z12-T112+1)
192   IF T1  0 LIGHT(4,1)
194   GOTO 178
1000  REM
1001  REM   SUBROUTINE OF CALCULATIONS
1002  REM
1004  LET E3=E
1006  LET T3=T1
1008  GOSUB 1102
1010  LET S1=S
1012  REM   CALCULATE FIRST AND SECOND SCATTER ENERGIE
1014  LET E1=E/(1+E*(1-T1))
1016  LET E2=E1/(1+E1*(1-T2))
1018  LET E3=E1
1020  LET T3 T2
1022  GOSUB 1102
1024  LET S2 S
1026  REM   SET THE SIZE OF THE BINS  2.5 KEV
1028  LET J INT(E2*511/2 5)
1030  LET H[J]  H[J] +S1*S2
1032  RETURN
1100  REM   SUBROUTINE TO CALCULATE SIGMA
1102  LET A=1+T3 2
1104  LET B  1+E3*(1-T3)
1106  LET C=(0 1) 2
1108  LET D=1+C/(A*B)
1110  LET S=3 665*A*D/(16*P*B 2)
1112  RETURN
1200  REM   OUTPUT OF E VS. PROB
1202  FOR I=1 TO 4
1204  LIGHT(I,0)
1206  NEXT I
1210  LET M=0
1212  FOR F=1 TO 250
1214  IF H[F]  M LET M=H[F]
1216  NEXT F
1218  REM   NORMALIZE THE PLOT
1220  LET P2=8500/M
1222  PRINT
1224  PRINT
1226  PRINT "E2", "PROB"
1228  PRINT
1230  FOR M=1 TO 250
1232  BPLOT ((1000/15)+36*M,1000+H[M] *P2)
1234  IF H[M]=0 GOTO 1238
1236  PRINT 2.5*M,H[M]
1238  CNTRL (1,0)
1240  NEXT M
1242  CNTRL (0,0)
1244  BPLOT (0,0)
1246  END
```

COMPUTERIZED PROCEDURE FOR ACQUISITION, STORAGE, AND MANIPULATION OF TOPOGRAPHIC DATA FOR USE IN SYSTEM ANALYSIS PROBLEMS

Phillip L. Doiron, Sr.
V. E. LaGarde
Mobility and Environmental Systems Laboratory
U. S. Army Engineer Waterways Experiment Station
Vicksburg, Mississippi

ABSTRACT. The U. S. Army Engineer Waterways Experiment Station has developed procedures for obtaining topographic data in a form suitable for use in general system analysis problems. Topographic data are considered at two levels of detail: microgeometry and macrogeometry. Microgeometry deals with the portion of the ground surface that occurs between the elevation contours and must be acquired by conducting on-site topographic surveys; whereas, macrogeometry deals with the topographic data as defined by the contours as shown on U. S. Geological Survey (USGS) maps.

The first step in the automated procedure is to obtain a data array (grid) of equally spaced discrete elevations from either a set of randomly located (i.e. field-measured) elevation points or the USGS map (i.e. contour data) for the particular site or area of interest. This gridded array of elevation points is referred to as a digital topographic model (DTM). Once the model has been established, it is used as input to various general computer programs to provide data in various forms for use. The type of data that can be generated from the DTM include elevation profiles, two-dimensional perspectives, contour maps, intervisibility maps (i.e. regions visible and in defilade from a point observer), and slopes. Parameter data in various forms can be specified on an areal basis or along a line or direction, such as the slope of the ground along a prescribed azimuth.

An example is presented for applying the model to a system analysis problem related to the determination of the effects of ground surface terrains on the vulnerability of different types of military targets.

1 INTRODUCTION. A computerized procedure, developed at the U. S. Army Engineer Waterways Experiment Station (WES) for generating topographic data in a form that is useful for a wide class of system analysis problems, has emerged from research sponsored by various Department of Defense agencies. The development motivation came primarily from the need for quantitative topographic data to be used in mathematical descriptions of materiel-terrain interactions for designers, test engineers, and tacticians; and in describing operations-terrain interactions for tactical and strategic planners and construction engineers. The procedure, known as the digital topographic model, embodies well-known, tested mathematical methods, and has been used by various researchers in studies at the WES over the past several years.

2. THE DIGITAL TOPOGRAPHIC MODEL PROCEDURE. A flow chart for the digital topographic model procedure is shown in fig. 1. Only three of the

many uses of the digital topographic model are shown in this figure, since they are the outputs needed for the example of its use to be presented later in this paper.

The digital topographic model procedure was designed to handle any level of detail in the input data and to produce output of corresponding detail. Its results can also be used as input to other more comprehensive models. The topographic model procedure can accept input data from a variety of sources, such as contour maps, field surveys, and aerial stereophoto interpretations. Of these, data from contour maps have been used most extensively, chiefly because contour maps are available covering wide areas of the world. Attention will be focused herein only on field survey data and contour maps. The procedures for using the model are described in the following paragraphs.

3. MICROGEOMETRY AND MACROGEOMETRY. The difference between microgeometry and macrogeometry is one of spatial resolution in defining the ground surface. Whether the elevation data for a surface are described as "micro" or "macro" is almost entirely arbitrary.

Microgeometry data must be acquired by on-site surveys. A stylized presentation of one method of surveying microgeometry data in the field is given in fig. 2. When the survey is performed in a grid format, as illustrated, elevations are read at the positions of a regular grid cast on the terrain. The data can also be surveyed in a non-gridded fashion. In this procedure the elevations are measured at topographic breaks in slope, which tends to produce a more-or-less randomized pattern of points.

Field survey data are placed on data forms in the field, the forms go directly to card punch operators, and the data are placed on computer punch-cards and computer processed. When the data are obtained by the grid method, the only function of computer processing is to convert the field record into an appropriately structured computer file of xyz coordinates. However, when the field data are obtained by a non-grid method, the random data points are processed by means of an elevation grid array program that yields xyz coordinates located on a regular grid. The program is described later in this paper.

For this paper, macrogeometry is defined as measurements taken from contour maps with a contour interval of 5 meters or greater. According to this definition, the topographic data as expressed by elevation contours on U. S. Geological Survey maps yield macrogeometry data.

The input data for the macrogeometry portion of the system is extracted from the maps by the use of the digitizer equipment shown in fig. 3. The positions of elevation data points, namely the contour lines, are entered automatically onto the magnetic tape by depressing a button on the digitizing "tracker" or "cursor". Other data are entered onto the tape through the keyboard. The magnetic tape containing the data is then immediately available for computer operations, or it can be stored for later use. Since

contour map data are analogous to ungridded data, these data are transformed, as the microgeometry ungridded data are, into grid data format through the use of the elevation grid array program.

4. ELEVATION GRID ARRAY PROGRAM. The elevation grid array program is used to convert the data obtained in the field or form topographic maps into a digital format for use on the computers. There are three interpolative procedures in the program. The topographic structure and spatial distribution of data points define which procedure should be used in a specific situation. These three procedures are: (a) inverse distance squared fit; (b) modified inverse distance squared fit; and (c) linear fit.

For the inverse distance squared fit (see fig. 4), the closest data point to the grid position in each of the four quadrants about the position is used in determining the elevation at that grid position. The procedure applies an inverse distance squared weighting factor to each of these four data points. The weighting factor has the property of giving the most weight to that data point closest to the grid position. W_j is the weighting factor associated with the j^{th} data point, d_j is the distance measured from the grid position, k is the number of data points used in the fit procedure, Z_j is the elevation value of the j^{th} data point, and Z is the interpolated elevation value at the grid position.

The modified inverse distance squared fit is similar to the inverse distance squared fit. However, instead of using one point per quadrant to determine the elevation, it uses a constant number of data points that are closest to the grid position. The number of data points, while not restricted, is normally between 4 and 40. This procedure has the capability of giving more weight to a cluster of data points near the grid position, and provides a smoother, more average surface than does the inverse distance squared fit, because all points chosen are used regardless of the quadrant in which they are located.

For the linear fit (see fig. 5), the closest data point to the grid position in each of the four quadrants about the position is used in determining the elevation at that grid position. However, unlike the two previous fits, the linear fit does not use a weighting factor. Rather it involves three linear interpolations to finally produce the elevation value at the grid position. The first interpolation is performed to calculate an elevation value along a line segment connecting the data point in the first quadrant with the data point in the fourth quadrant. $X_{1,4}$ is the interpolated elevation value along the line segment at the point when it intersects the X axis. Y_4 is the Y coordinate of the data point in the fourth quadrant, Z_4 is the elevation value of the data point in the fourth quadrant, Y_1 is the Y coordinate of the data point in the first quadrant, and Z_1 is the elevation value of the data point in the first quadrant. The location of the interpolated elevation values along the line segment is denoted by $X'_{1,4}$, and X_1 and X_4 are the X coordinates of the data points in the first and fourth quadrants, respectively. The second step involves interpolating an elevation value along a line segment

connecting the data point in the second quadrant with the data point in the third quadrant. The values in these two equations are the same as for the previous equations, except that the data points in the second and third quadrants are connected by a line segment with a second interpolated elevation value located on the segment at $X'_{2,3}$. The final step involves a line segment connecting $X'_{1,4}$ with $X'_{2,3}$, and a final interpolation of elevation at the grid position.

The output of the three surface fitting procedures is an elevation grid array. This array consists of interpolated elevation values located at all grid positions within the data site.

5. APPLICATION TO A SYSTEM ANALYSIS PROBLEM. As stated previously, the digital topographic model can be used for many types of computer calculations. One recently completed WES study in which results of the model were required as input is presented to illustrate its flexibility.

This study was concerned with the natural shielding provided by the terrain to targets on a road, such as vehicles and personnel, from fragmenting munitions bursting at various distances from the road and at various heights above ground. A set of analytical procedures was developed for calculating the amount of shielding that the topographic surface provided.

A grid was compiled for each data site as discussed above, and various automated techniques were used to portray and analyze the site.

Perspectives drawn by the computer (see fig. 6 for an example) were produced with the digital topographic model to give an overall view of the site. This technique is also very useful in the detection of errors in the model.

Profiles were extracted from the topographic model to show the shape of the terrain along particular paths. A comparison of a profile extracted from the model and a profile surveyed in the field (fig. 7) illustrates how closely the model depicts the terrain.

Contour maps were produced (fig. 8) to show a conventional view of the site. This is also another technique to find errors in the model.

A further computation was made with the model for this particular study. A 90-cm-tall target was located at the center of a site. Visibility was calculated and converted into a computer plot (fig. 9) that shows the portions of the site from which the 90-cm-height region of the target would be exposed and vulnerable to ground-bursting munitions.

The actual model developed to calculate the shielding places a target on the road at specified coordinates. Munition bursts are then placed at various ranges and various heights above the ground from the target, and the rays from the burst to the target are checked against the ground-surface configuration to see if the rays reach the target.

6. CONCLUSION. The digital topographic model developed by the procedures described has shown great flexibility in meeting requirements imposed by different projects over the past several years. These projects have ranged from fish population studies, where output of the model was used as input to a calculation of water depths, water surface area, and the amount of terrain covered by the water at different stages, to tank-antitank warfare, when the model was used as input to calculations of areas where tanks would be visible in the study areas.

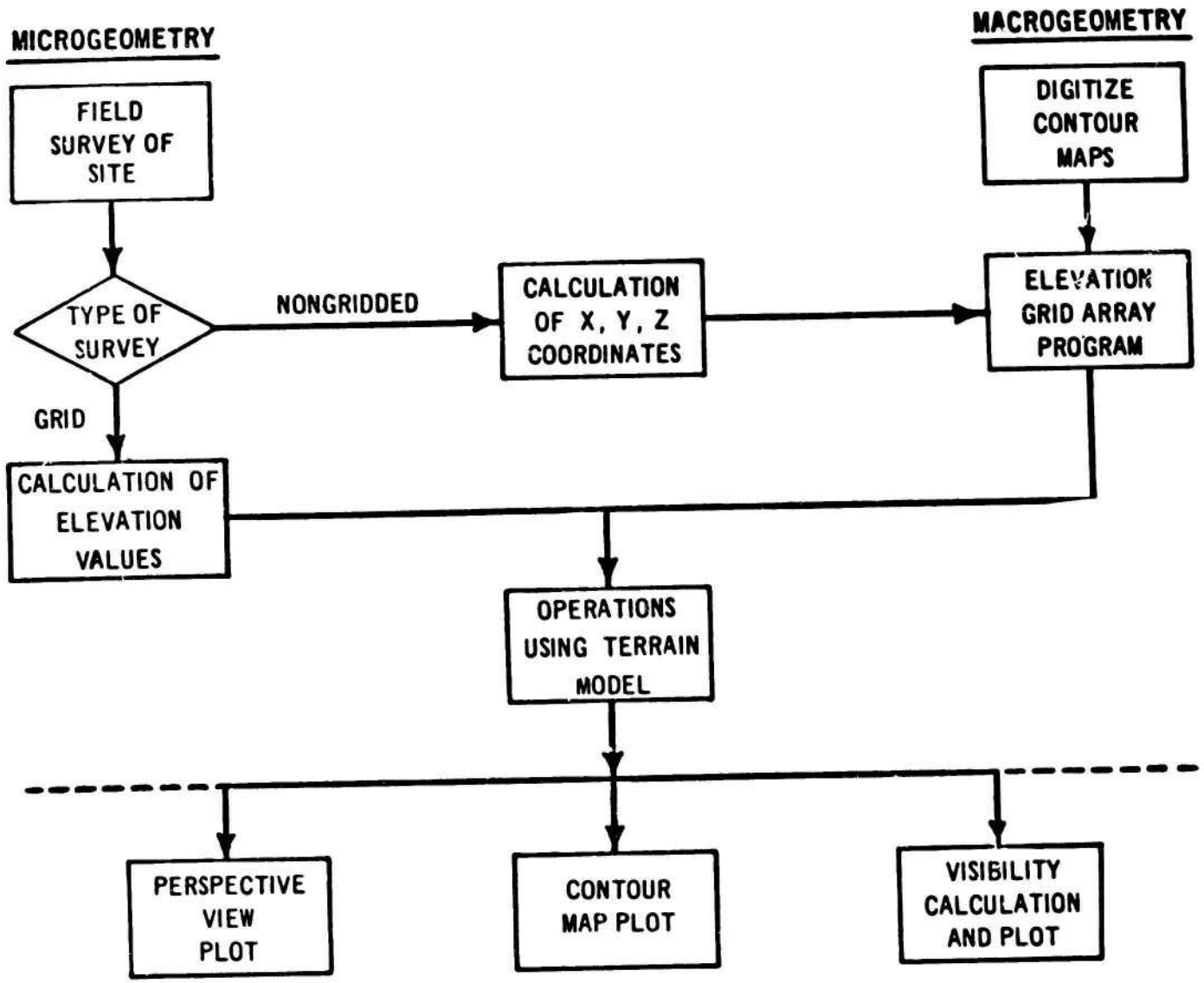


Fig. 1. Synopsis of steps in digital topographic model procedure

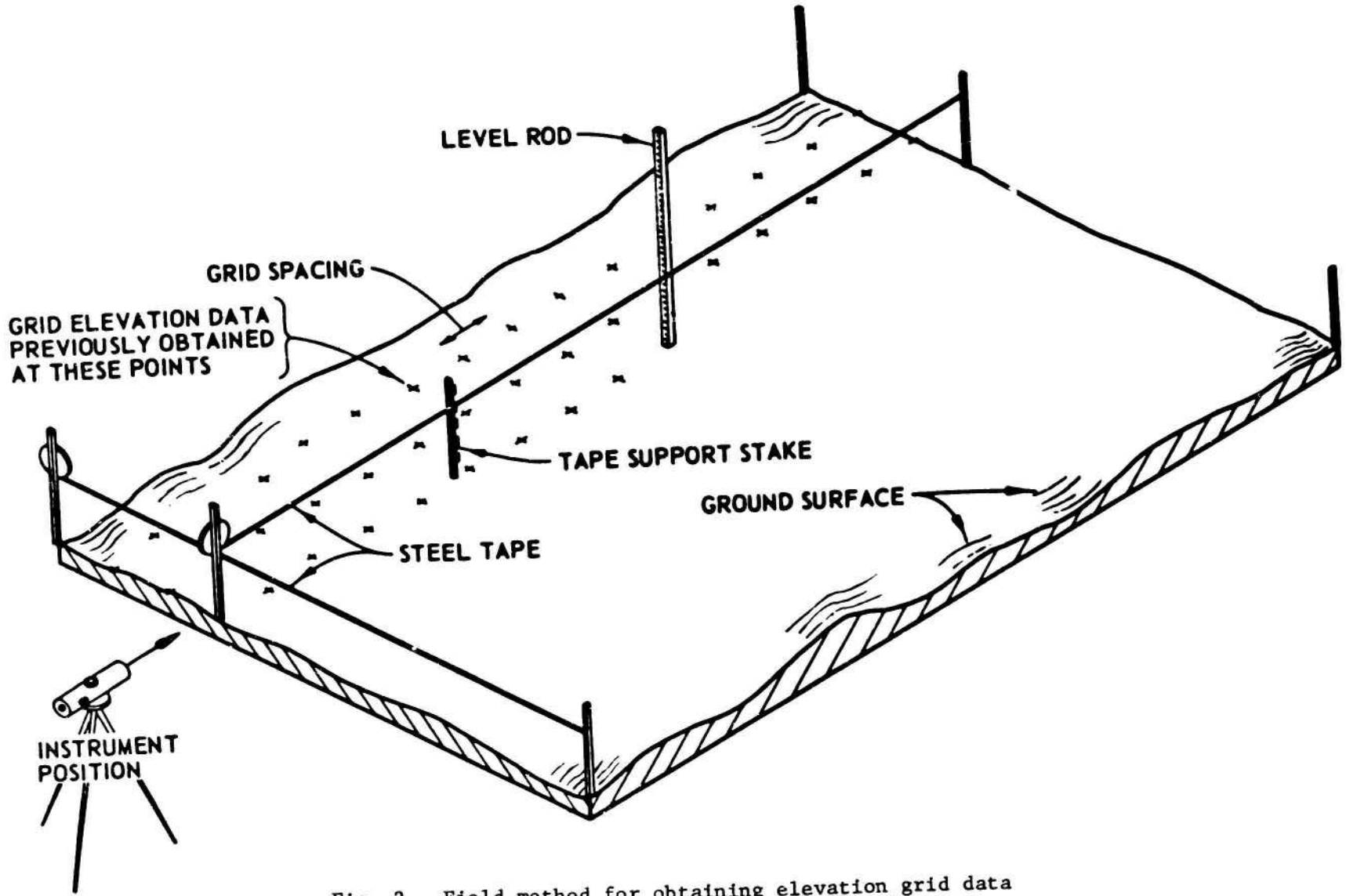
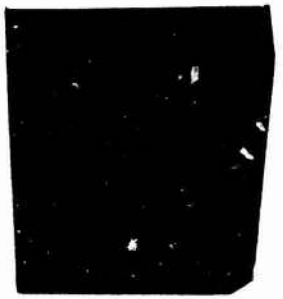


Fig. 2. Field method for obtaining elevation grid data

TAPE



KEYBOARD

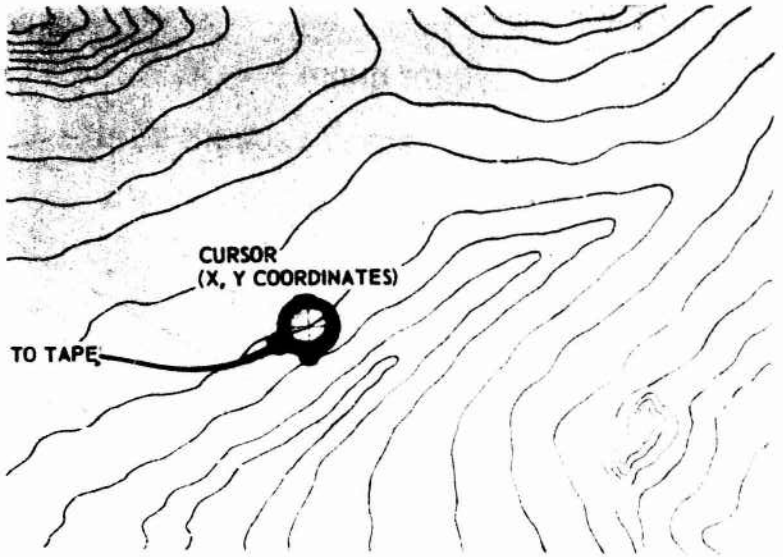
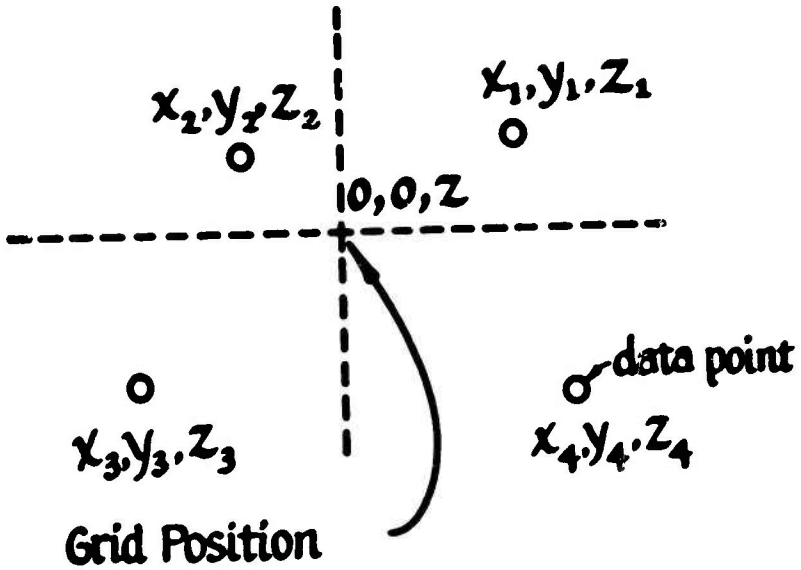


Fig. 3. System for obtaining digitized data



$$w_j = \frac{d_j^{-2}}{\sum_{i=1}^{k} d_i^{-2}}$$

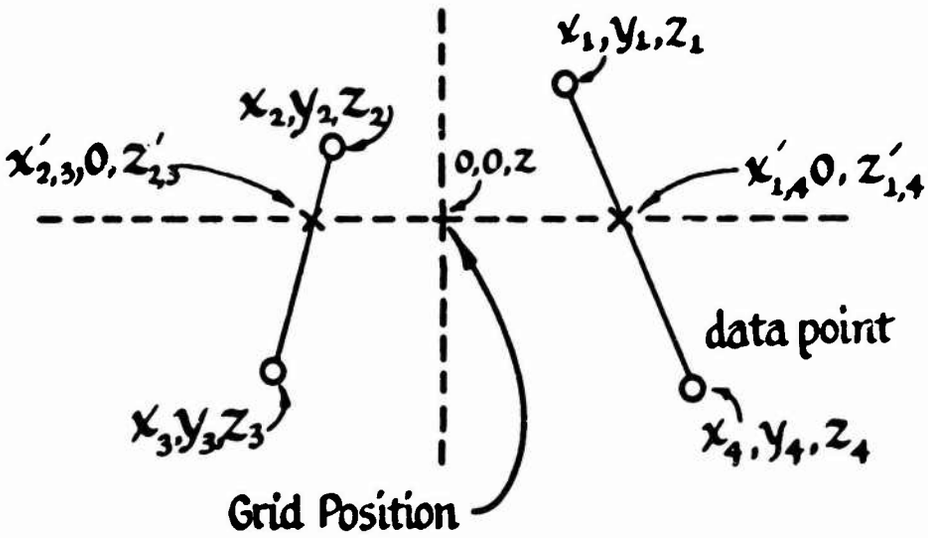
$$d_j^2 = x_j^2 + y_j^2$$

a. Calculation of Weighting Factors

$$z = \sum_{j=1}^{k} w_j z_j$$

b. Calculation of Elevation Value at Grid Position

Fig. 4. Inverse distance squared fit method



$$z'_{1,4} = \frac{y_4 z_1 - y_1 z_4}{y_4 - y_1}$$

$$x'_{1,4} = \frac{y_4 x_1 - y_1 x_4}{y_4 - y_1}$$

$$z'_{2,3} = \frac{y_3 z_2 - y_2 z_3}{y_3 - y_2}$$

$$x'_{2,3} = \frac{y_3 x_2 - y_2 x_3}{y_3 - y_2}$$

$$z = \frac{z'_{1,4} x'_{2,3} - z'_{2,3} x'_{1,4}}{x'_{2,3} - x'_{1,4}}$$

Fig. 5. Linear fit method

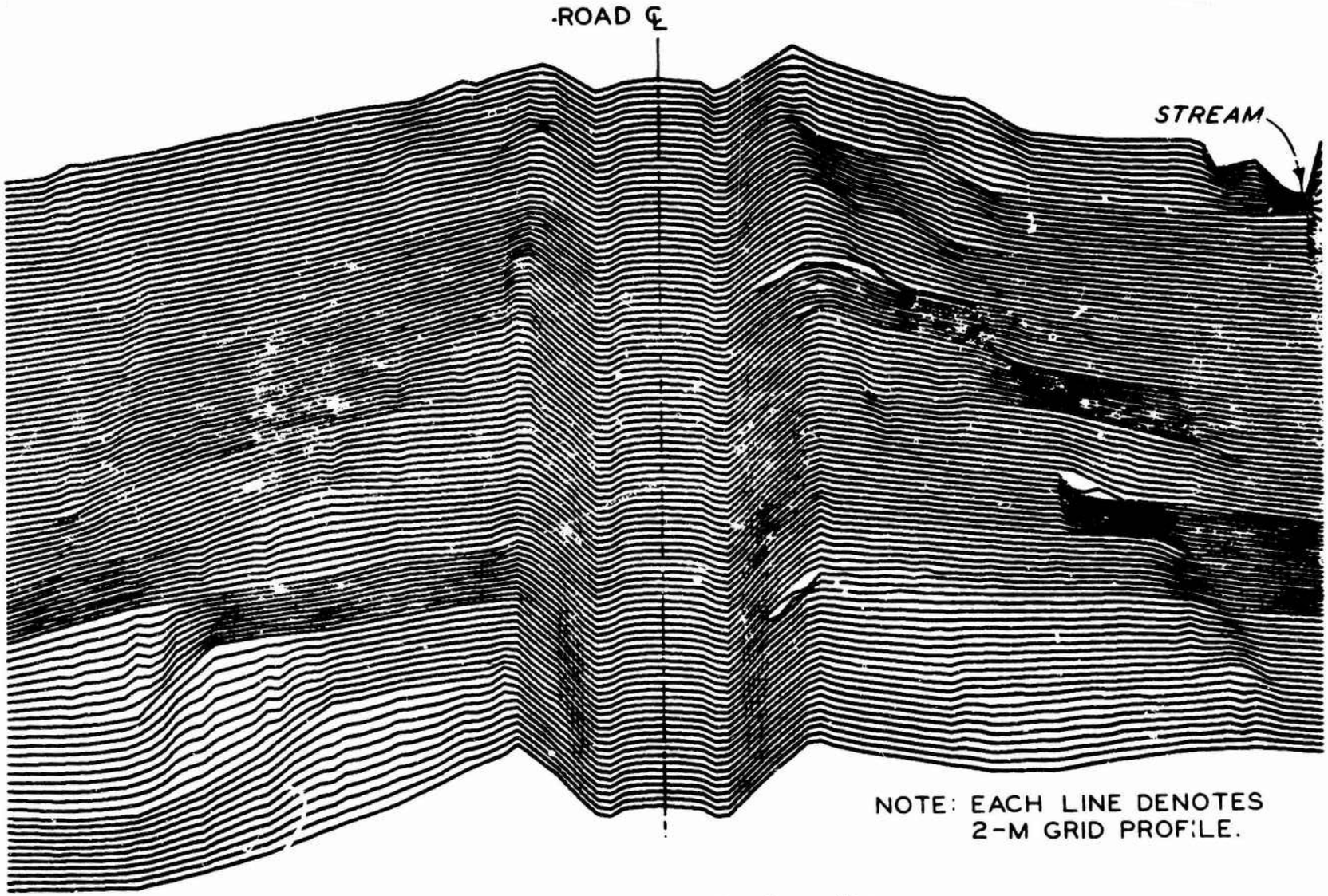


Fig. 6. Perspective of a data site

Fig. 7. Profiles extracted from digital topographic model and surveyed in the field

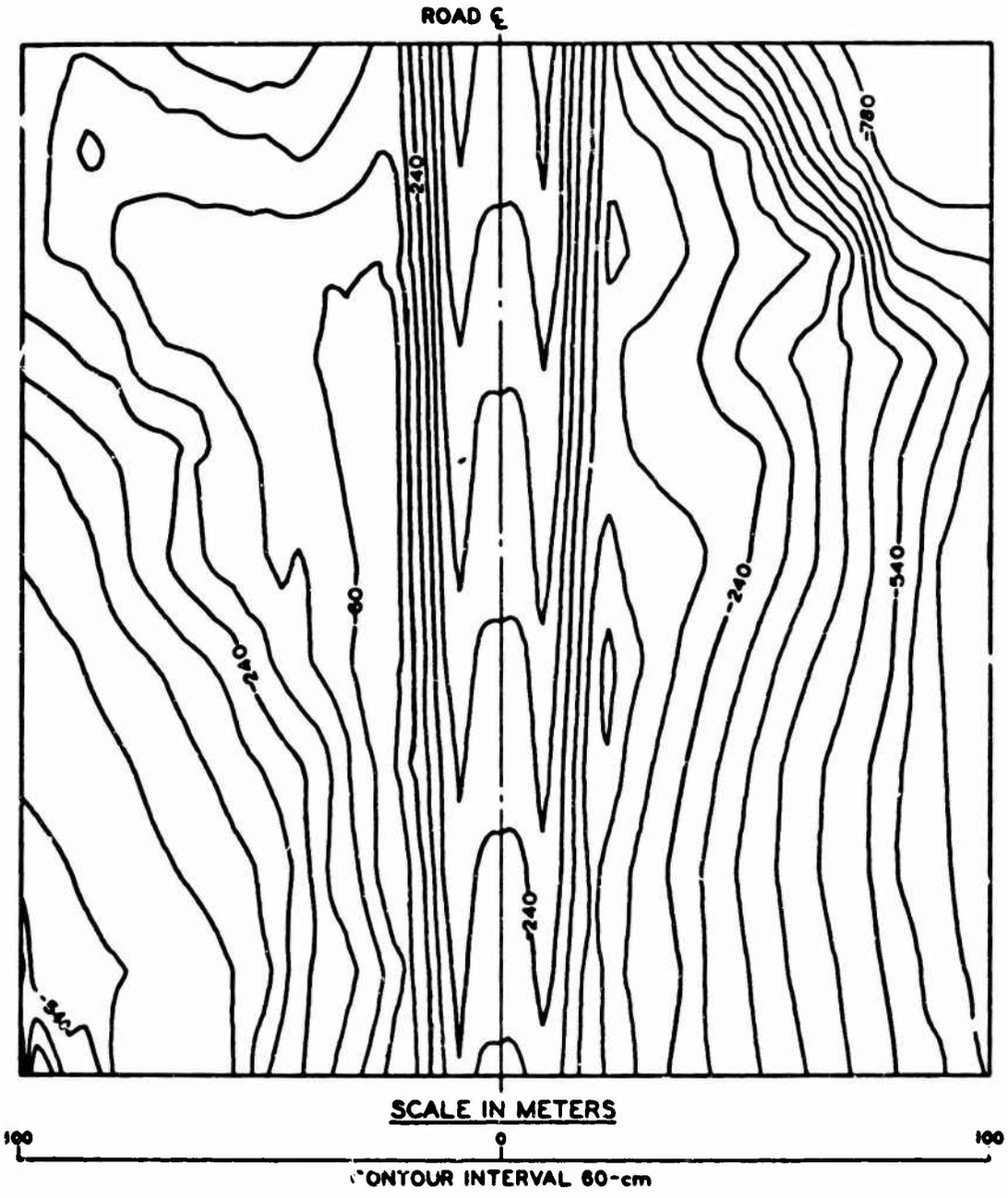


Fig. 8. Contour map of a data site

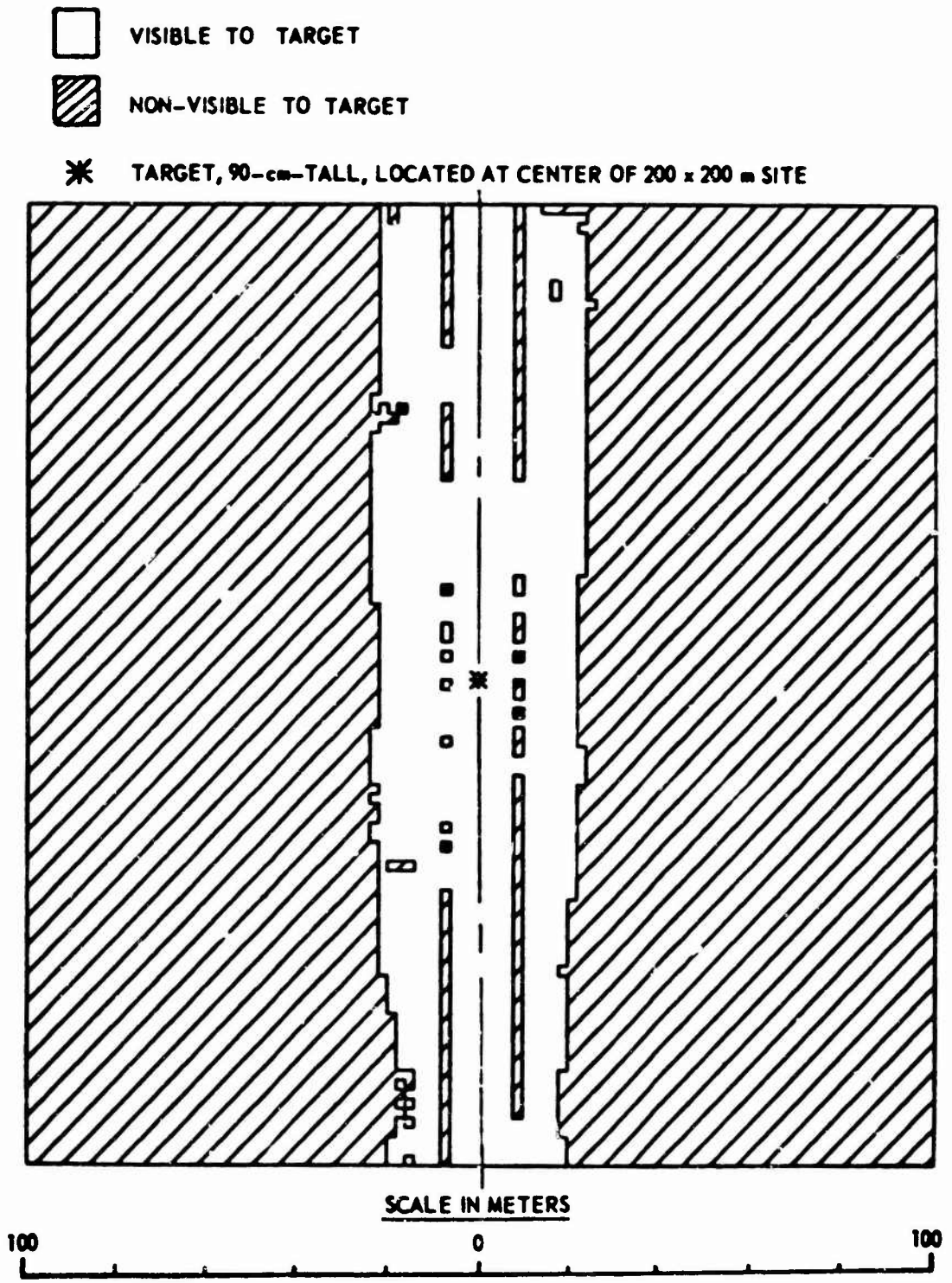


Fig. 9. Computer plot of visible and non-visible regions from a target height of 90-cm

COMPUTATION OF SMOOTH CONTOURS OVER ARBITRARY PLANAR REGIONS

Richard J. Bair
Benet Weapons Laboratory
Watervliet Arsenal
Watervliet, New York 12189

ABSTRACT. A procedure has been developed to generate smooth labelled contours from discrete data points of the form Z=Z(X,Y). The procedure implements a piecewise doubly cubic spline approximation to generate a smooth surface from the data points and uses traditional root-finding techniques to track the desired contour across the surface in small increments in the X and Y directions. The process, implemented in a FORTRAN program, can be used to track contours defined on regions which are arbitrary polygons. The technique uses boundary information and a "directed ray" to determine if a particular point is interior or exterior to the region. If interior, the contour is tracked.

The ability to track contours on regions defined as arbitrary polygons makes the technique particularly well suited to the output from finite element analyses.

METHOD. Frequently, the need arises, in scientific and engineering applications to make an orderly assessment of the data produced from a problem with large amounts of output. It would be desirable to provide the scientist or engineer with a tool whereby he could rapidly determine the form of the region. Contour lines for the region when outputted to a graphics device, provide such a tool.

There are two principal steps in the process of generating smooth contours from discrete data points. First, it is necessary to generate a representative surface from the data. Second, the value of the contour in question must be tracked across the surface.

Several methods exist for handling the surface generation problem. Linear interpolation, quadric interpolation and bicubic splines are three types of methods in use. The smoothness of the contours is directly related to the surface generation technique used. Since the contour tracking routine is implemented in a FORTRAN computer program, all that is required is that the surface be computed by a FORTRAN FUNCTION subprogram.

Contours may be tracked across regions approximated as arbitrary polygons. The approximation of all planar regions by some polygon or union of polygons allows for the most general of cases. The problem of tracking the contour in question is facilitated by overlaying a rectangular grid of cells on the region of interest.

X_{low}, X_{high}, Y_{low}, Y_{high} define the area of in. rest. This area is subdivided into a grid as in Figure 1.

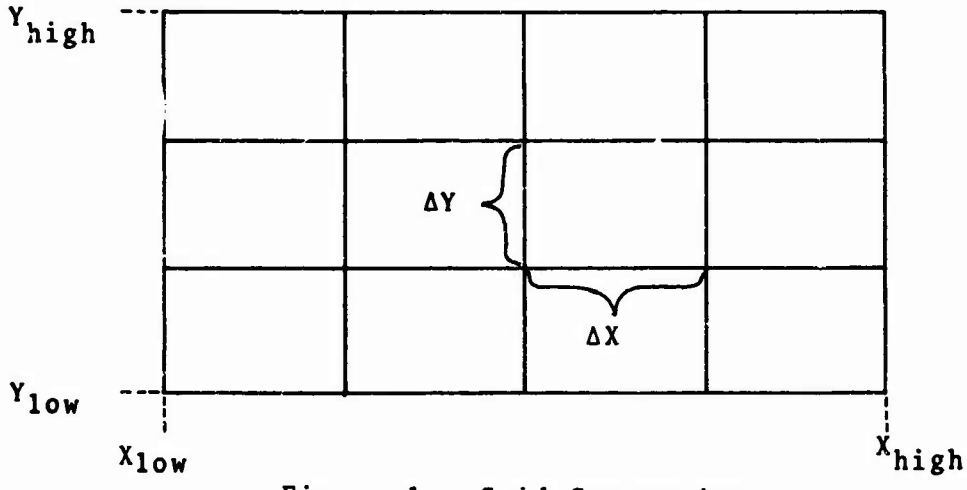


Figure 1. Grid Generation

If the contour value falls within the range of the maximum and minimum functional values at the corners of a cell, each of the cell sides is checked to determine which one it intersects.

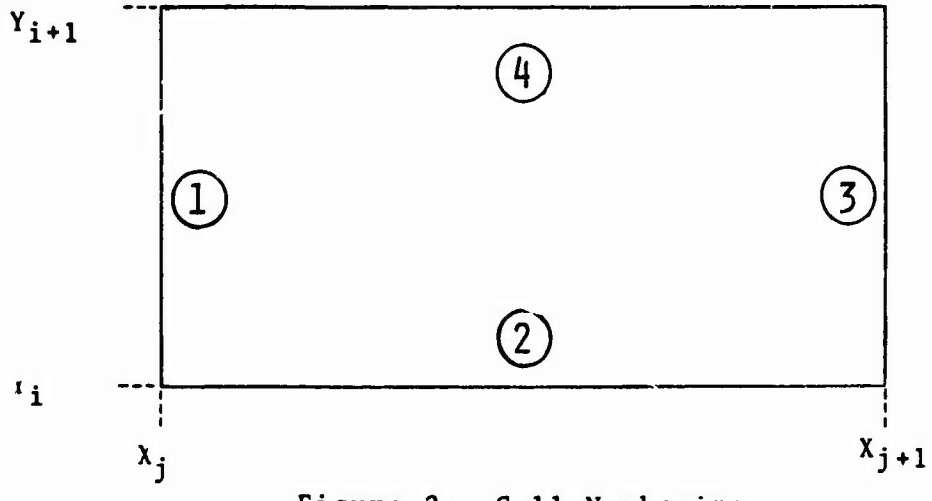


Figure 2. Cell Numbering

Once the cell sides have been checked, it is necessary to actually track the contour across the cell in small increments. The process is illustrated in Figure 7

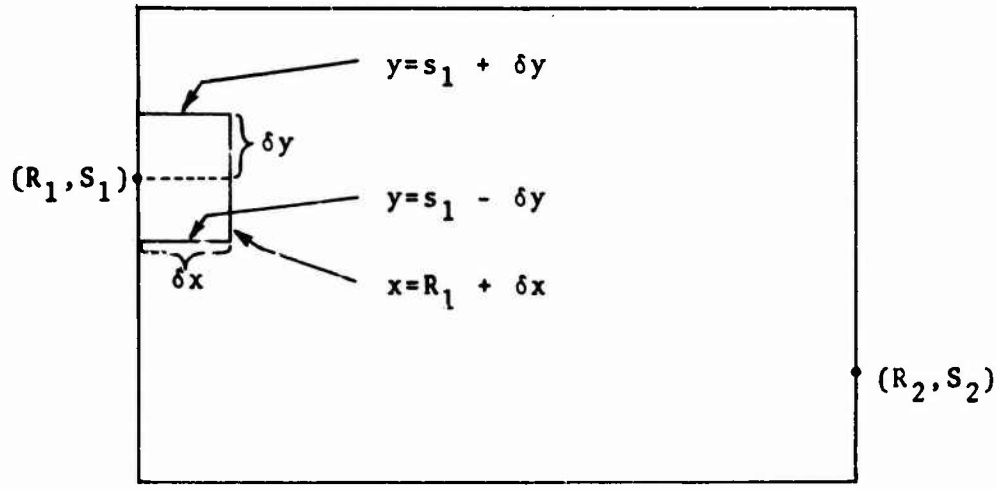


Figure 3. Tracking Within a Cell

Points (R1,S1) and (R_2,S_2) are found via the preliminary root-finding process. Starting at (R_1,S_1) advance in the positive X direction a distance δX. Along the line

$X=R_1 + \delta X$ examine for roots of the function:

$Z_v(y)=C$ where $y\ (s_1 - \delta y, s_1 + \delta y)$

If no root is found check in two other directions for a root. First, along the line $Y=s_1 + \delta y$ examine for roots of:

$Z_h(X)=C$

Second, along the line $y=s_1 - \delta y$ for roots of:

$Z_h(X)=C$

The distances δx and δy are a function of the distance between consecutive points found on the cell sides.

$$\delta x = \frac{|R_2 - R_1|}{10.} \qquad \delta y = \frac{|S_2 - S_1|}{10.}$$

These increments could be made smaller, though computation time would then increase.

An important advantage in the reduction of the problem to a root-finding problem is that each of the points on the contour is obtained within the same relative error.

Once a root is found a line is drawn on the graphics device to the new point from the former one. The tracking procedure continues to perform a local exploration of the cell until the newest point is within a small distance $\varepsilon>0$ of (R_2,S_2). Then a line is drawn between those two points and the process repeated in the next cell. Depending upon whether the last contour line was obtained via a move to the west, east, north or south, the next attempted move will be in the same direction.

In cases where all rectangular grids are not interior to the region of interest, as in the arbitrary regions, it is necessary to determine whether a particular polygonal pair of (X,Y) coordinates is interior to the region of interest. If it is, a bright vector is drawn on the graphics device from the last interior point to the current one. If not, a dark vector is drawn until an interior point is found. This particular technique makes use of a "directed ray".[1]

From the point in question, examine the intersections of some directed ray from that point with the sides of the approximating polygon. If there are an even number of side intersections, the point is exterior to the region; if odd, it is interior to the region.

Figures 4 and 5 show representative results of contours tracked on irregularly shaped regions obtained through finite element analyses. In each case, the approximating surface was obtained through the use of piecewise doubly cubic splines. Figure 4 is a plot of circumferential stresses obtained from a NASTRAN analysis of the spindle in the XM199E8 design of the 155mm howitzer. The plot reveals the smoothness of contours tracked over a region with a polygonal boundary. Figure 5 depicts stresses obtained in the static simulation of tooling during operation of high pressure equipment. This plot shows a nonrectangular region with multiple interior boundaries defining areas of differing element densities and material properties.

Since this paper is intended to be an extension to an earlier report[2] dealing with tracking contours on rectangular regions, additional details of the process and a listing of the computer program may be found in that source.

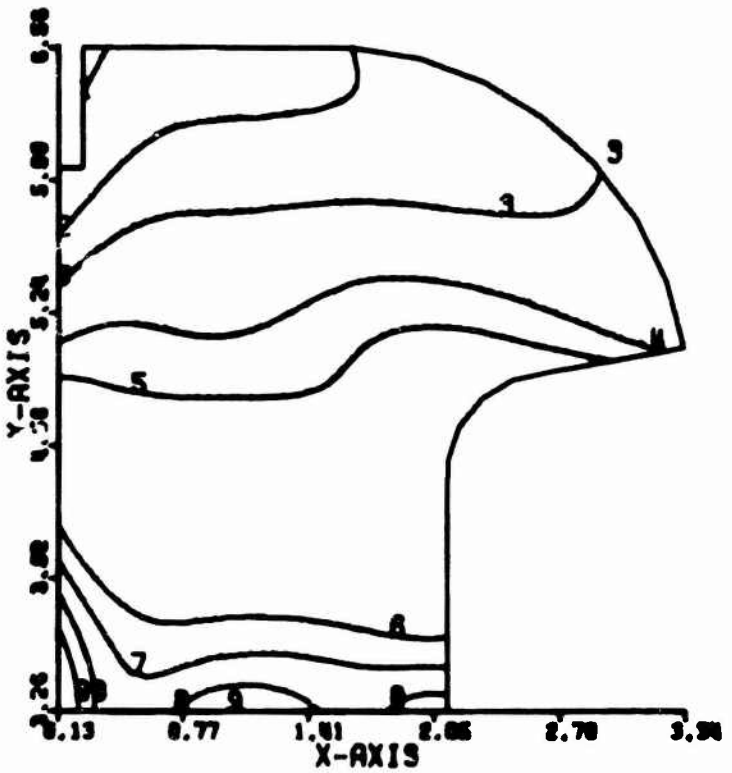


Figure 4. Circumferential spindle stresses.

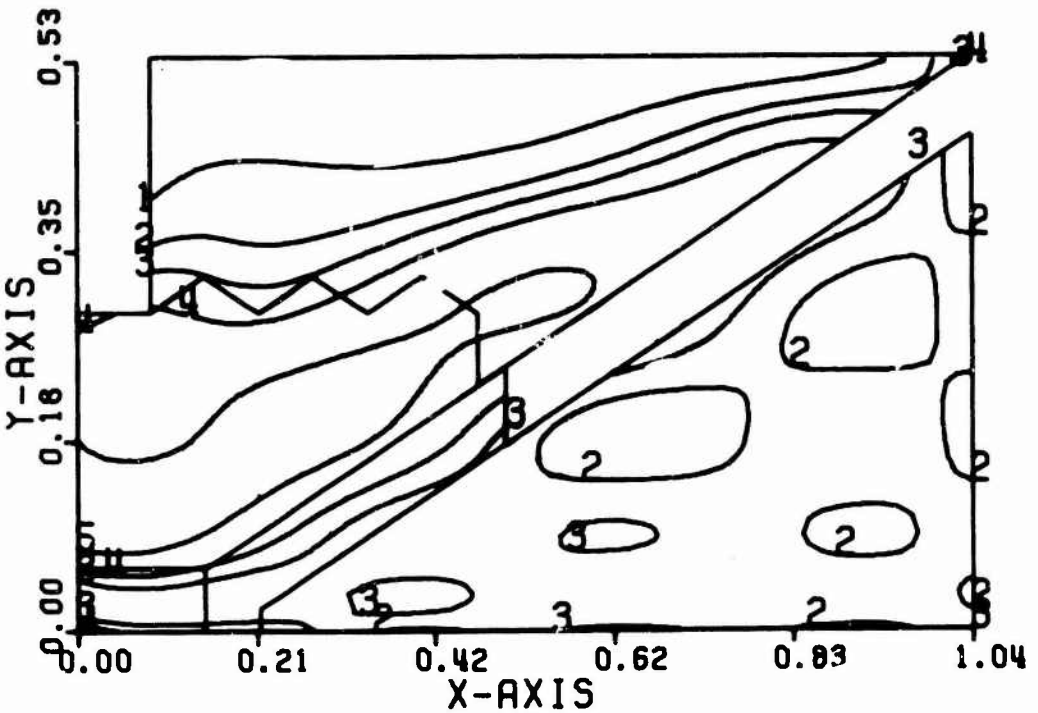


Figure 5. Tooling equipment stresses.

REFERENCES

1. Jacobsen, J. D., Geometric Relationships for Retrieval of Geographic Information, IBM Systems Journal, Vol. 7, Number 3, 1968, pp 331-341.

2. Bair, R. J., Computation of Smooth Contours from Non-Uniform Data, Watervliet Arsenal Technical Report, WVT-7265, Dec 1972, Watervliet, New York.

THE TWO-STREAM INSTABILITY STUDIED WITH FOUR ONE-DIMENSIONAL PLASMA SIMULATION MODELS

David L. Brown
Fire Control and Engineering Directorate
Frankford Arsenal
Philadelphia, Pennsylvania

ABSTRACT

Four one-dimensional plasma simulation models have been compared with regard to the electrostatic two-stream instability. The primary reason for making these comparisons was to determine the extent to which physical results depend on numerical method for a problem in which collective effects dominate. Previously, Lewis, Sykes and Wesson compared these four simulation models using a stable double-streaming situation as a test problem. In that case the comparisons were with regard to collisional effects, energy conservation, and momentum conservation; however, because a stable test problem was used, only tentative conclusions could be drawn as to the comparison among the models when they are applied to a problem in which collective effects dominate. We have applied the models to compute the evolution of a two-steam instability, and compared the time-dependence of the electric energy as determined by each of the models. The models are characterized by the representation of the electric potential, and by how the electric field is computed from the potential; both linear and quadratic splines are used to represent the potential or field The major result of our comparisons is that the evolution of the electric energy of a two-stream unstable plasma does not depend strongly on the choice of model. There is a much stronger dependence on the random numbers that are chosen to represent the initial distribution function in phase space.

This paper is to appear in full in the Journal of Computational Physics.

A CALIBRATION PROCEDURE FOR A BALLISTICALLY EMPLACED ACOUSTIC BEARING SENSOR ARRAY

Kenneth J. Dean
Systems Division
Countermine/Counter Intrusion Department
U. S. Army Mobility Equipment Research
and Development Center
Fort Belvoir, Virginia

ABSTRACT. A non-linear multiple regression model is developed as a calibration procedure for a ballistically emplaced acoustic bearing sensor array. The calibrated baseline values are used as an intermediate step in predicting relative target position location. The tracking accuracy obtained from the calibrated array is sufficient to adjust artillery fire on moving targets. Sensor position adjustments are determined by resectioning based on the geometrical variations observed in the triangulated intercepts about an estimated target path. Simultaneous target bearings and the approximate size of the array about which the sensors are assumed to be randomly placed are required as input data. Constraints on target motion are not imposed except as required to provide simultaneous bearings by numerical interpolation techniques. Validation of the mathematical model and computer code is based on a linear target path through a 500 meter array. The accuracy of the regression and its zone of convergence is investigated for selected parameters.

FOREWORD. The methods and techniques used in this report were adopted from procedures employed by the USAF and NASA in the calibration of range tracking stations and the calculation of trajectory data for vehicles launched at the Atlantic Missile Range. The book edited by Dr. Ernest H. Ehling "Range Instrumentation", Prentice-Hall, 1967 was most helpful. Without the guidance and confidence provided by chapters written by Dr. Ehling "Optical Instrumentation" and Dr. Rudolf Burns "Doppler Systems", it is doubtful that the numerical verification of this model would have been attempted.

Appreciation is also extended to Mr. Herb Thompson, Special Projects Division, who originally described the problem and developed the technique by which the absolute location of the array can be determined.

1. INTRODUCTION. A target is assumed to be tracked by acoustic bearing sensors as it moves through a rectangular array. If its relative motion along the path is smooth and continuous, such that simultaneous

bearings can be determined by interpolation, its relative position to the array can be computed by triangulation between sensor pairs. Unless the bearings and geometry of the array are precisely known, the six values (four items taken two at a time yield six combinations corresponding to the sides and diagonals of the array) will disagree.

Geometrical distortions in the array, caused by sensor placement errors, introduce systematic shifts in computed target positions that are dependent upon the relative target sensor geometry. This is illustrated in Figure 1 by taking bearings measured from the sensor and projecting them from their assumed locations. Random noise in the bearing data compounds the problem. Since the geometry is constantly changing for a moving target, errors in predicted target position will also vary at different points along the path. This variation in the ambiguity of the solutions then implicity contain the baseline errors.

Describing this relationship mathematically by the partial derivatives of target position with respect to the baselines, a set of linear equations can be written in terms of sensor position errors. The partials are sometimes referred to as the GEDOP numbers (geometrical dilution of position) and relate the variation in triangulated position due to a unit change in baseline. Since the baseline errors are fixed at the time of sensor placement, the set of unique equations existing at each simultaneous bearing can be combined and solved by the least squares method. This yields the "best" set of baseline corrections required to minimize the squared sum of deviations between predicted target locations. Geometrically, this corresponds to adusting sensor positions to their triangulated intercepts at various points across the array are coincident. In a similar fashion, the effect of random noise in the tracking data can be reduced by regressing on the individual bearings. A mathematical derivation of these procedures are presented in Appendix A.

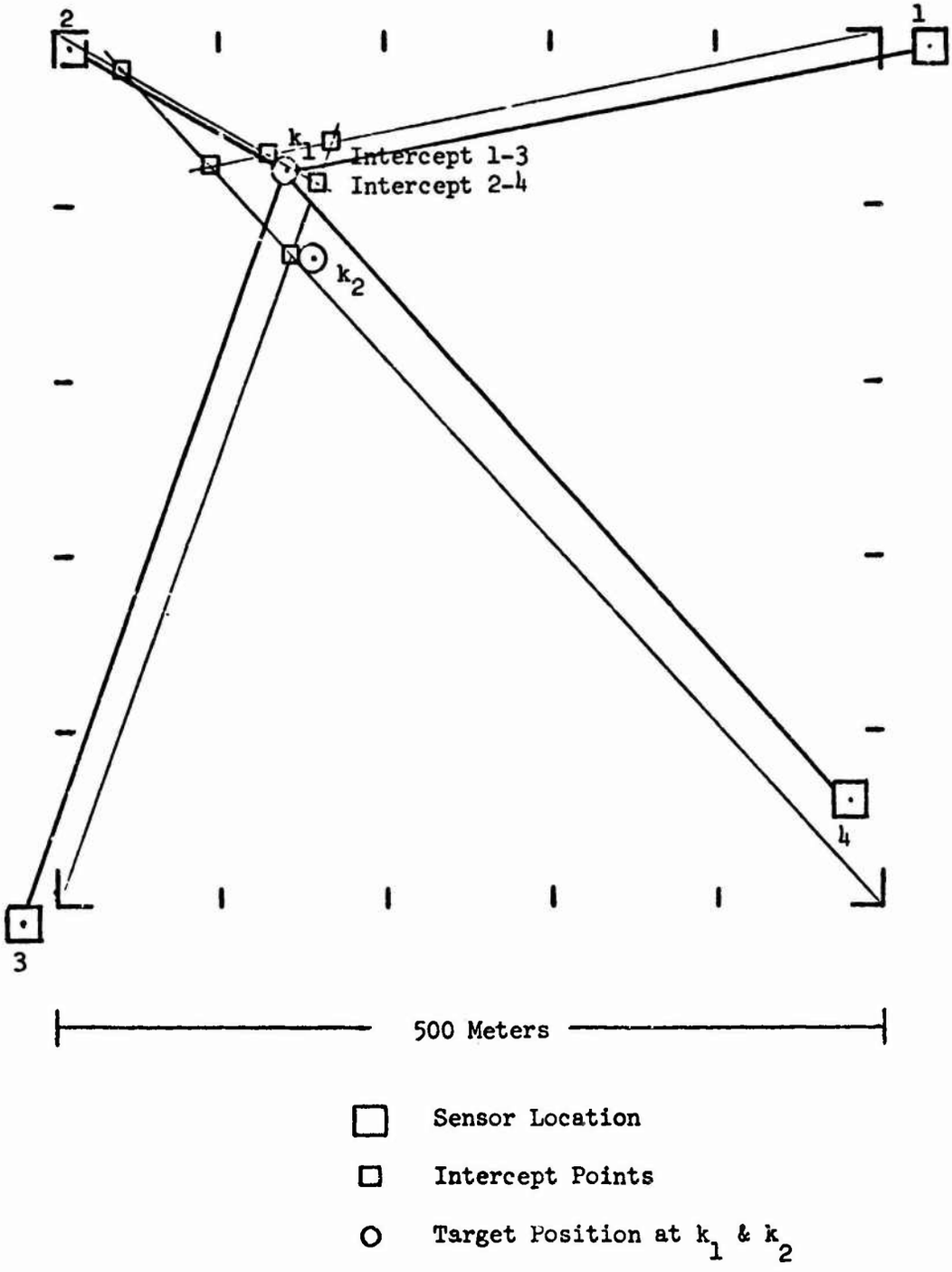
Although the intercepts or predicted target points can be forced to converge, the position that this convergence occurs about is not necessarily the spatial location of the actual target path. The discrepancy between the two occurs because distance is not measured directly by the sensor and must be estimated on the basis of the initial dimensions of the array. Some of the parameters affecting the accuracy of this approximation are investigated in the following section.

Assuming now that simultaneous bearings can be provided by data pre-processing, the relative target position can be determined by regression. The absolute location of the array is still limited by the basic accuracy of the delivery system. Thompson[1] has proposed an innovative approach by which this uncertainty can be eliminated. In essence, the sensor array is used in lieu of a forward observer to adjust artillery

---

[1]Thompson, Herbert H., "Ubiquitous Target Position Location," CM/CI Technical Note, Draft.

Figure 1, Position by Triangulation From a 500 Meter Baseline



fire. The calibration compliments this approach by improving the relative accuracy of the array, thereby reducing the number of rounds required to arrive on target. Once registration is achieve, the procedure need not be repeated, except for periodic updating and compensation of secondary effects.

2. DISCUSSION. A test case representing a 500 meter array was constructed to verify the regression model and to conduct a sensitivity analysis of selected parameters. The sensors were randomly placed about the aimpoints with a 25 meter CEP delivery accuracy. Actual placement errors are shown below and in Figure II with the sensors numbered counterclockwise

SENSOR PLACEMENT ERRORS (METERS)

Sensor I 40, - 10	Sensor III -30, -10
Sensor II 10, - 10	Sensor IV -20, -60

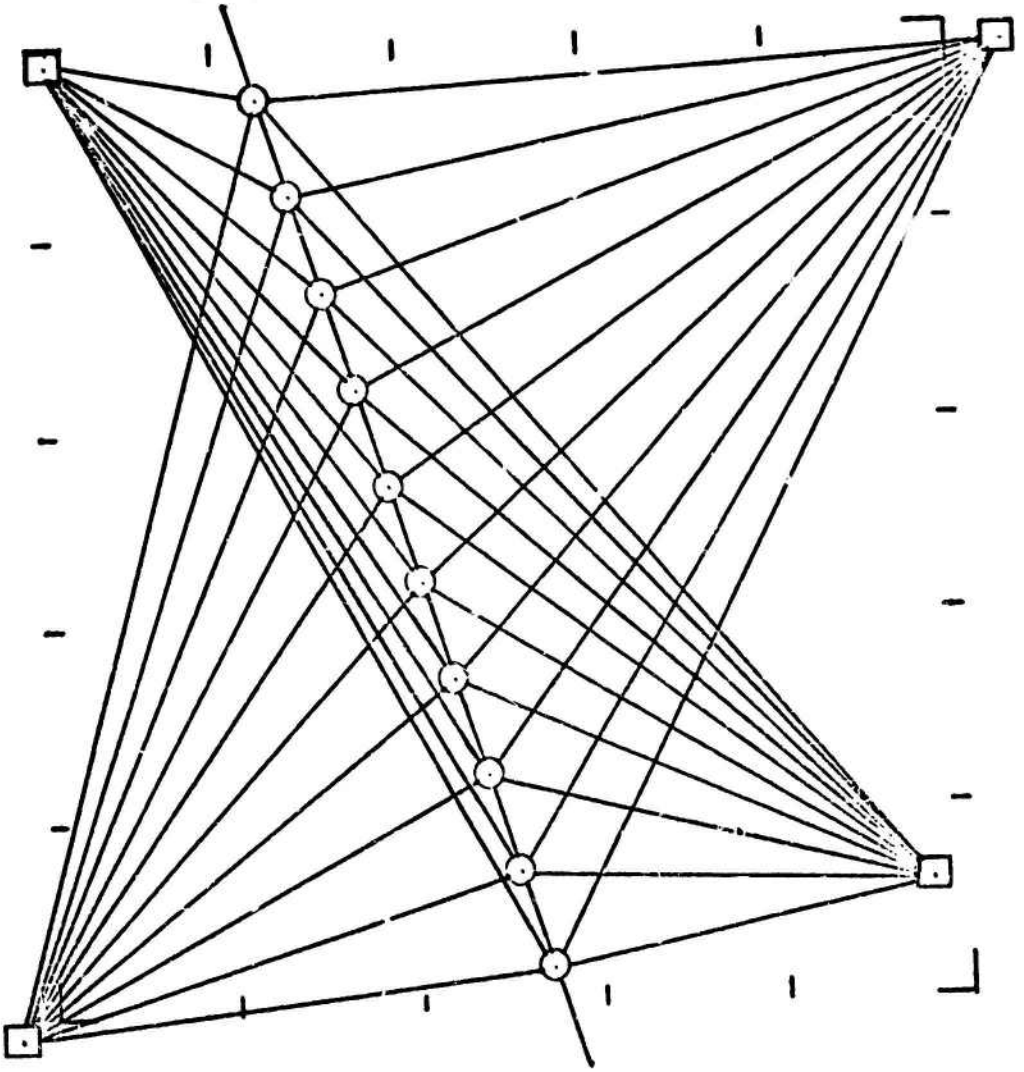
beginning at the upper right hand corner. Data point spacing along the track approximates a 10 mile per hour target velocity and a 10 second sample interval.

Simultaneous bearings were measured directly from the figure to an accuracy of 0.2 degrees. This is considered representative of typical smoothing-interpolation techniques and avoids the added complexities ties involved with data pre-processing that would otherwise be required. Likewise, a curvilinear path and varying target velocity was not modeled since their effects can only be introduced through the pre-processing and are highly technique dependent.

The ambiguity in predicted target position is shown in Figure III for sensor pairs 1-4 and 2-3 at their initial positions where the origin of the reference coordinate system has been translated to the stationary sensor. Geometrical variation in tracking accuracy in this illustration is most pronounced for the 1-4 sensor pair. Initially, the error is minor due to the 4th intercept vector passing very near the sensor's true position; as the target orientation deviates from this axis, an increasingly larger component of the placement error is added to the intercept. This can best be described in terms of parallax where the orientation to a fixed object is changed by a displacement in the point of observation. Similar variations also exist in the other sensor pairs. Extremely large errors can be observed if the bearings are by chance taken near the baseline crossings where the intercept equations become degenerate.

Approximating the target path by the average of sensor pairs 1-4 and 2-3, correction components of (-25.1, 2.1), (-55.4, -25.1) and (-42.8, 48.4) were computed for the 2nd, 3rd, and 4th sensors respectively.

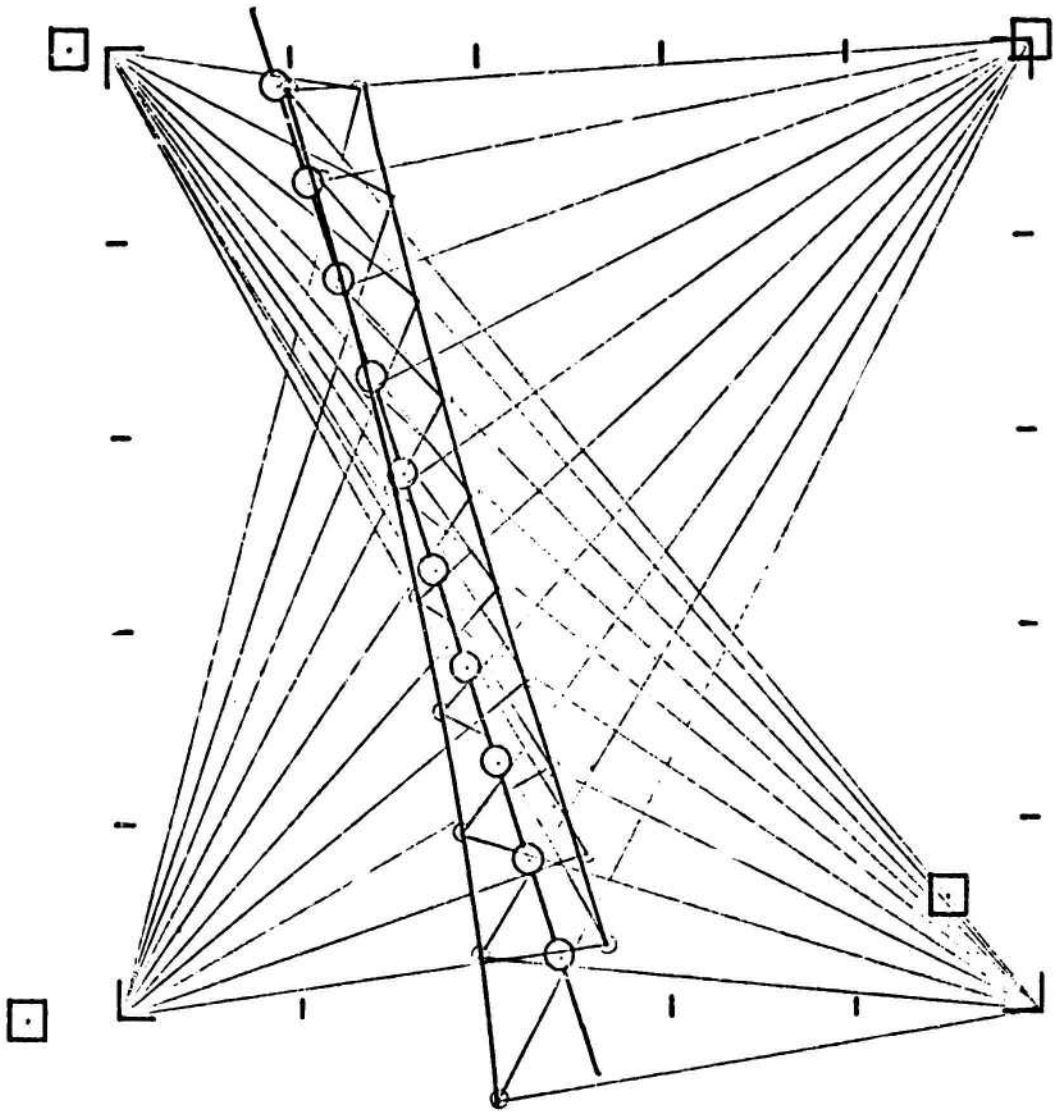
Figure II; Sensor Locations and Simultaneous Bearings
500 Meter Array and 47.4 M RMS Sensor Error



○ Target Position

□ Sensor Location

Figure III, Predicted Target Track Before Adjustment
47.4 Meter RMS Sensor Location Error



Sensor Location

Target Position

Predicted Position

Comparing to the original errors of (-30,0), (-50,0) and -50,-60), the corrections effectively reduce the initial 47.43 meter RMS sensor error to 10.4 meters. The improvement provided by this iteration is shown in Figure IV. Variations between the two predicted tracks which had previously averaged 32 meters have been reduced to 4 meters. Also, the misalignment between predicted and actual target heading has improved from 4.5 to 0.1 degrees although the value eventually settles out at -0.5 degrees.

Results of subsequent interations are given in Table I, in terms of goodness of fit criteria and tracking accuracy in position, velocity and azimuth. These are self explanatory with the possible exception of the geometrical ratio which expresses the similarity between the shapes of the computed and actual arrays. It is defined by the ratio of the quotient of the diagonals in the computed array, to the quotient of the actual diagonals. As the sides of the two arrays become parallel, the value of the geometrical ratio approaches one.

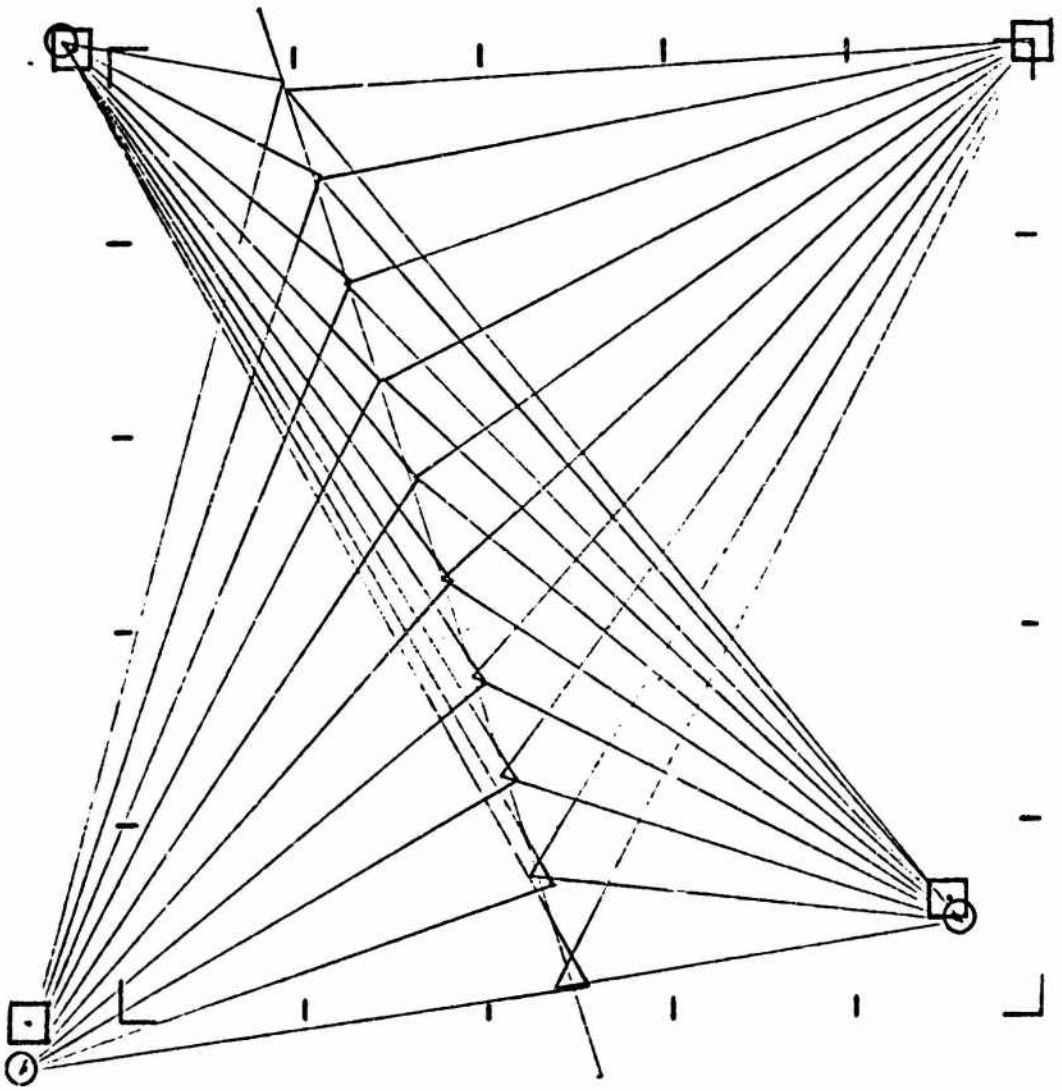
It is noted in Table 1, that the second iteration provided the best fit in terms of the position tracking error; however, since the standard deviation of the regression residuals is the only criteria available in actual practice to judge goodness of fit, the regression was continued until the differential threshold value (0.0005) was reached at the 7th iteration. By the completion of the 3rd iteration, 99% of the adjustment to sensor positions had been achieved. Sensor position continued to improve during the subsequent iterations. The remaining parameters remained unchanged or degraded slightly. An acceptable threshold level at which the regression should be terminated will require additional study.

Sensor placement error was selected as a primary parameter for sensitivity analysis since it was considered to be a principal error source. The evaluation was conducted with the base case presented in Figure II by moving the assumed sensor locations at which the regression was initiated. Irregular arrays and incremental baseline changes to the original 500 meter array were considered; i.e., 510, 490, 480, etc. Individual bearings were corrected to 0.01 degrees to minimize the influence of other error sources.

A total of 15 cases were run representing placement errors varying from 20 to 69.9 meters RMS, Figure V. Mean position tracking errors ranged from 0.3 meters to 11.5 meters for the irregular shaped arrays with little apparent relationship to initial sensor placement error; however, a definite trend is apparent for the incremental baselines.

Since the magnitude of the original placement accuracy must be discounted, the variations in incremental baselines suggest random bias as a possible error source in the regression.

Figure IV, Predicted Target Track at 1st Iteration
10.4 Meter RMS Sensor Location Error



Sensor Location

Predicted Location

TABLE 1

CONVERGENCE OF BASE CASE WITH 10 POINT SMOOTHED DATA 500 METER ARRAY AND 47.4 METER RMS SENSOR LOCATION ERROR

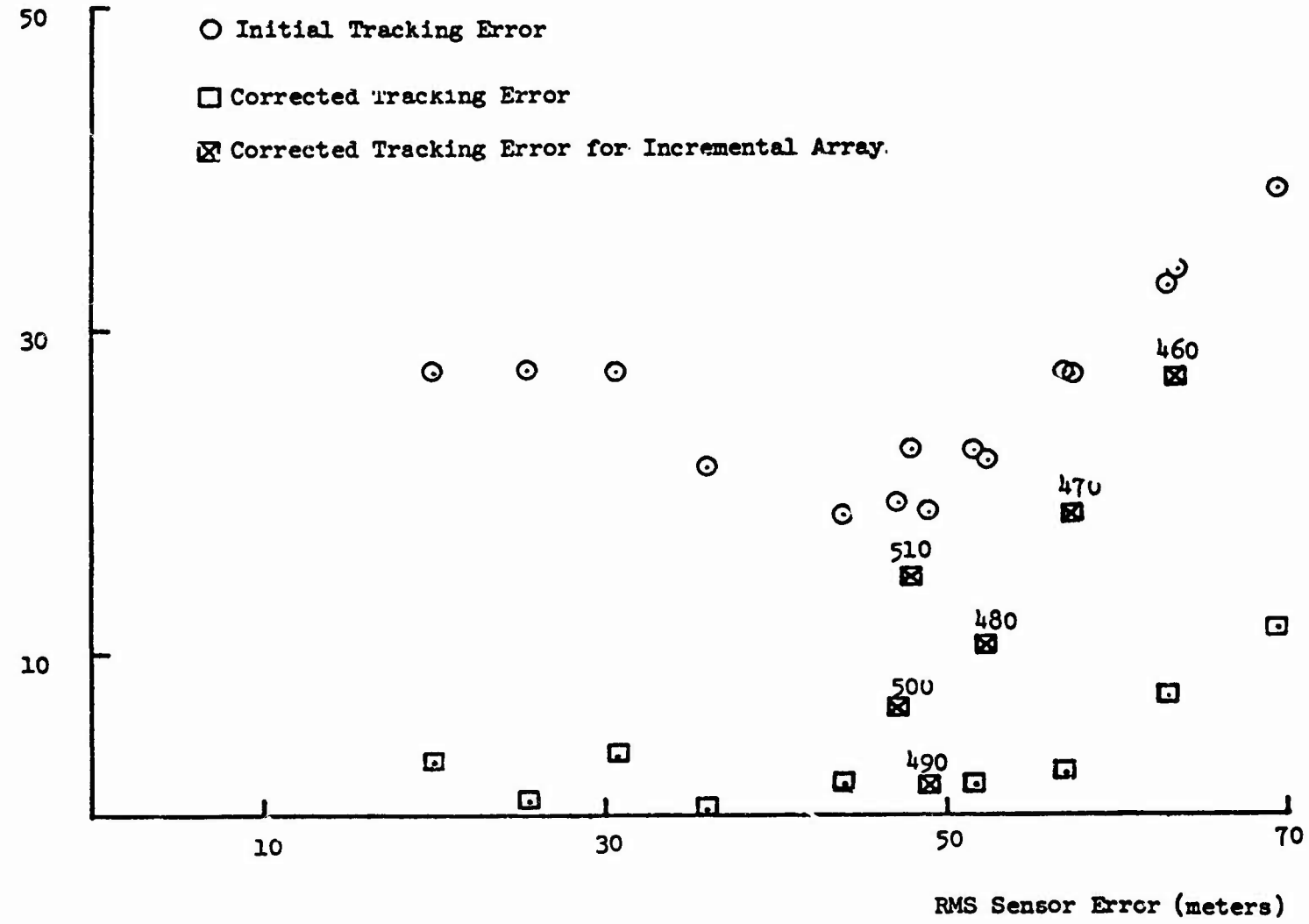
NUMBER OF ITERATION	REGR. RESID (M)	SENSOR POSITION ERROR		TARGET TRACKING ERRORS					
				POSITION (M)		VELOCITY (M/SEC)		AZIMUTH (DEG)	
	STD DEV	RMS (M)	GEO RATIO	MEAN	STD DEV	MEAN	STD DEV	MEAN	STD DEV
INITIAL	542.66	47.43	1.1545	16.07	13.99	0.49	0.29	4.45	1.56
1	5.70	10.42	0.9998	4.24	5.01	.16	.17	.11	1.07
2	2.46	7.52	.9885	3.56	3.16	.09	.14	−.37	.91
3	2.07	6.74	.9862	3.61	2.65	.08	.13	−.48	.83
4	2.03	6.53	.9855	3.63	2.52	.07	.13	−.52	.80
5	2.02	6.46	.9853	3.63	2.48	.07	.13	−.53	.80
6	2.02	6.43	.9852	3.63	2.47	.07	.13	−.54	.79
7	2.02	6.42	.9852	3.62	2.47	.07	.13	−.54	.79

TABLE II

TRACKING ACCURACY vs. SENSOR PLACEMENT ERROR WITH BEARINGS TO 0.01 DEGREES

SENSOR LOCATION ERROR METERS RMS	REGR. RESID (M) STO DEV	SENSOR POSITION ERROR RMS (M)	SENSOR POSITION ERROR GEO RATIO	TARGET TRACKING ERRORS POSITION (M) MEAN	POSITION (M) STO DEV	VELOCITY (M/SEC) MEAN	VELOCITY (M/SEC) STD DEV	AZIMUTH (DEG). MEAN	AZIMUTH (DEG). STO DEV
IRREGULAR ARRAYS									
20.0	.492	2.54	1.00048	3.09	1.21	.035	0.51	.015	.130
25.5	.494	0.68	1.00048	0.92	0.46	.007	.052	.015	.130
30.8	.491	3.03	1.00048	3.65	1.42	.042	051	.015	.130
36.1	.496	0.43	1.00048	0.29	0.36	.008	.052	.015	.130
44.0	.498	1.87	1.00048	1.98	0.87	.030	.052	.015	.130
51.7	.498	1.82	1.00048	1.91	0.84	.029	.052	.015	.130
57.0	.492	2.10	1.00049	2.56	1.02	.028	.052	.016	.130
63.1	.487	5.95	1.00048	7.02	2.71	.085	.051	.015	.130
69.6	.482	9.85	1.00048	11.50	4.44	.143	.050	.015	.130
SQUARE ARRAYS									
48.0 (510 M ARRAY)	.513	2.96	1.00048	14.75	5.76	.194	.054	.015	.130
47.3 (500 M ARRAY)	.503	5.71	1.00048	6.40	2.54	.087	.053	.015	.130
49.0 (490 M ARRAY)	.493	1.56	1.00048	1.95	0.79	.020	.052	.015	.130
52.4 (480 M ARRAY)	.483	8.80	1.00048	10.30	3.97	.127	.051	.015	.130
57.5 (470 M ARRAY)	.473	16.05	1.00048	18.65	7.20	.234	.050	.015	.130
63.7 (460 M ARRAY)	.463	2[illegible]1	1.00048	27.00	10.43	.341	.048	.015	.130

Figure V. Mean Position Tracking Accuracy vs. Initial Sensor Error



Computing random bias as the arithmetic sum of the differential base errors, Table III, and plotting it versus the mean tracking error confirms this as the primary error. Values deviating from the straight line function shown in Figure VI correspond to the irregular arrays. A specific cause for this secondary effect was not isolated. It probably represents a variation in the model's resolution as the criticality of individual sensor errors change in the initial geometry. The dependency of the regression model to random bias is inherent to the basic input data; i.e., target distance is not measured directly. Consequently, any array with the appropriate geometrical shape, regardless of its size, is sufficient to meet the angular requirements presented by the bearings. Since an infinite number of such arrays exists, the regression moves to the one nearest the initial starting point.

The effect of random bias on tracking accuracy is shown graphically in Figure VIII for the six incremental baselines evaluated. Basically, the predicated target locations are shifted to the left and downward for negative biases and to the right and upward for positive biases. Since this effect results in a constant offset, it must be distinguished from a random error in that subsequent predictions based on the same calibrated values will contain an equal or approximate bias.

Take the 460 meter calibrated array for example and suppose that an artillery shell lands 120 meters from the 7th target location as shown in Figure VIII, the calibrated array gives a value of 110 meters. A difference that is comparable to the standard deviation about the mean position error.

Three averaging functions were evaluated to determine the best in terms of position tracking accuracy: 1) the two condition case described above where the baseline of the sensor pairs parallel the target tracks; 2) an alternate 2 condition case where the baseline switches from a parallel to a transverse configuration relative to the target track between successive iterations; and 3) a 6 condition set using all possible pairs. The comparative results in position tracking accuracy, Table IV, indicated that the 2 condition case is generally preferred. This conclusion can also be inferred geometrically since the second two cases both involve one or more situations where the target may lie near the transverse baseline at which point the intercept equations become degenerate.

Additonal precision in the target approximation function can be provided by more sophisticated techniques if required. One such possible approach is commonly referred to as "Triangulation by Intersection".[2] It parallels the basic non-linear regression technique

---

[2] Ehling, Ernest H., "Range Instrumentation", Prentice-Hale, Inc., 1967, pg 98.

RANDOM BIAS IN SENSOR LOCATION ERROR

TABLE III

RMS SENSOR PLACEMENT ERROR (M)	RANDOM BIAS (M)	POSITION TRACKING ERROR (M)	
		MEAN	STD DEV
IRREGULAR ARRAYS			
20.0	– 81.7	– 3.09	1.21
25.5	– 49.8	– 0.92	0.46
30.8	– 28.6	– 3.65	1.42
36.1	6.8	0.29	0.36
44.0	– 23.0	– 1.98	0.87
51.7	– 57.3	– 1.91	0.84
57.0	– 75.4	– 2.56	1.02
63.1	– 93.9	– 7.02	2.71
69.9	–113.0	–11.50	4.44
SQUARE ARRAYS			
48.0 (510 M ARRAY)	–108.2	–14.75	5.76
47.0 (500 M ARRAY)	– 39.9	– 6.40	2.54
49.0 (490 M ARRAY)	28.4	1.95	0.79
52.4 (480 M ARRAY)	96.7	10.30	3.97
57.5 (470 M ARRAY)	164.9	18.65	7.20
63.7 (460 M ARRAY)	233.3	27.00	10.43

Figure VI. Position Tracking Errors vs. Random Bias

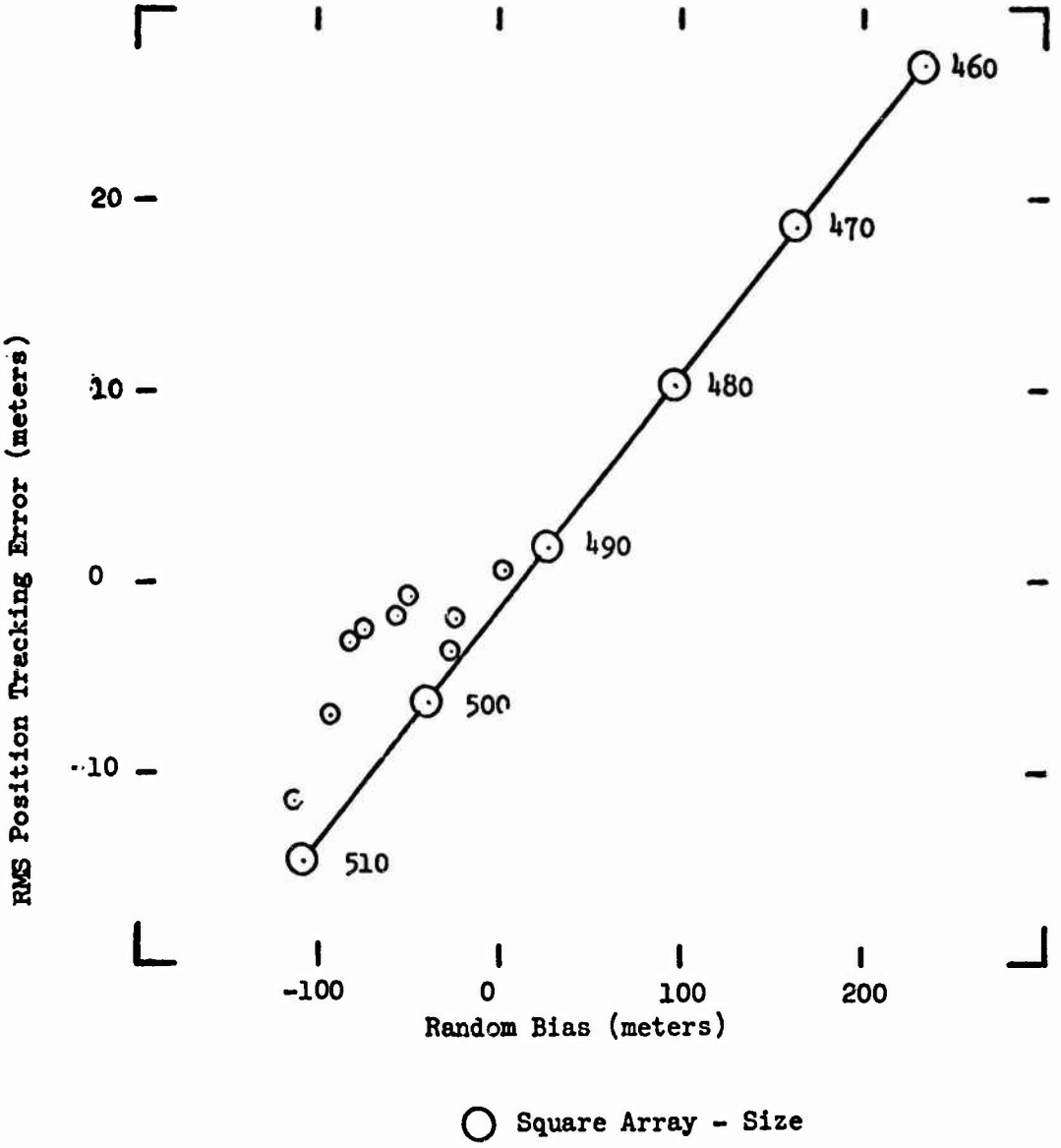
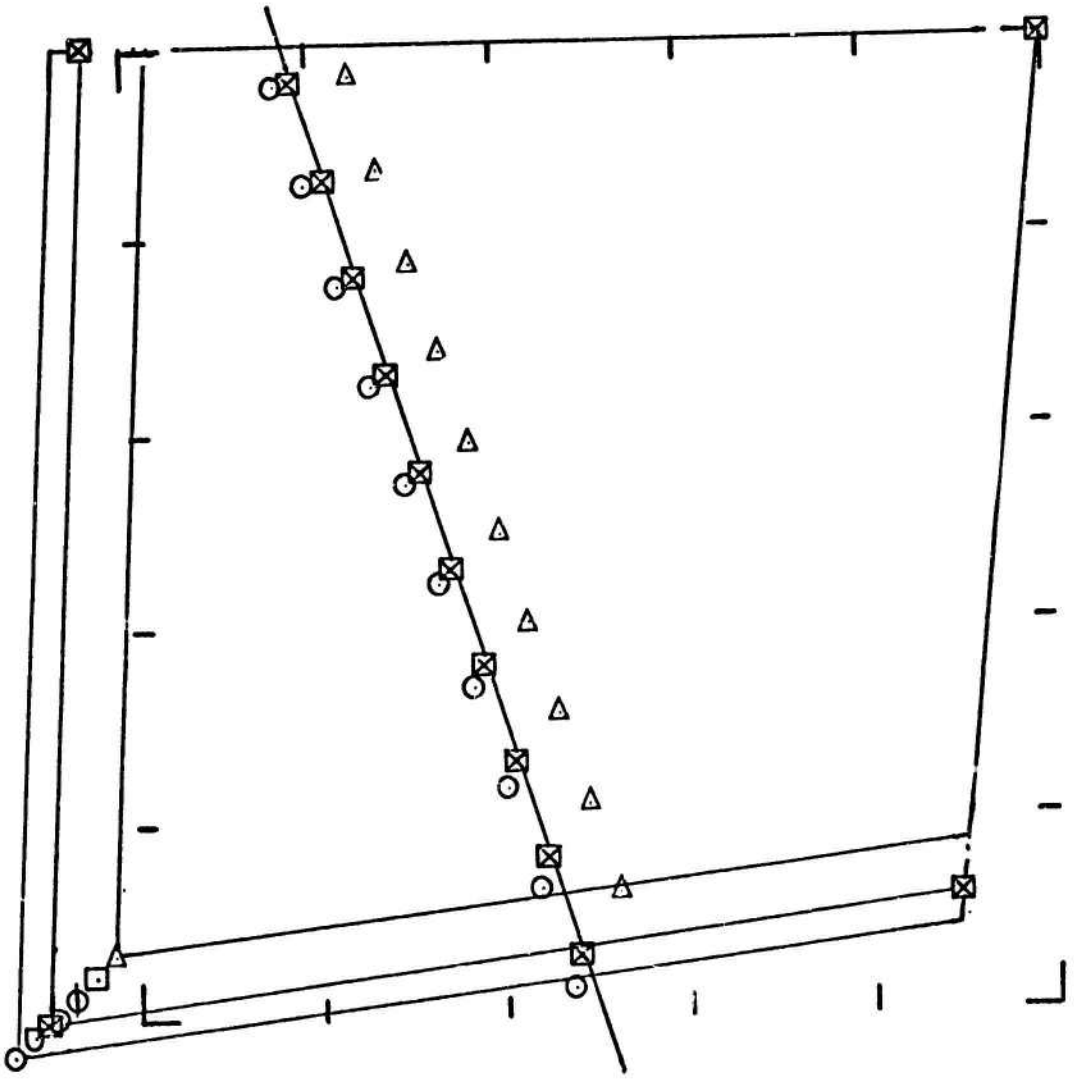


Figure VII, Calibrated Configurations For
Incremental Baseline Arrays

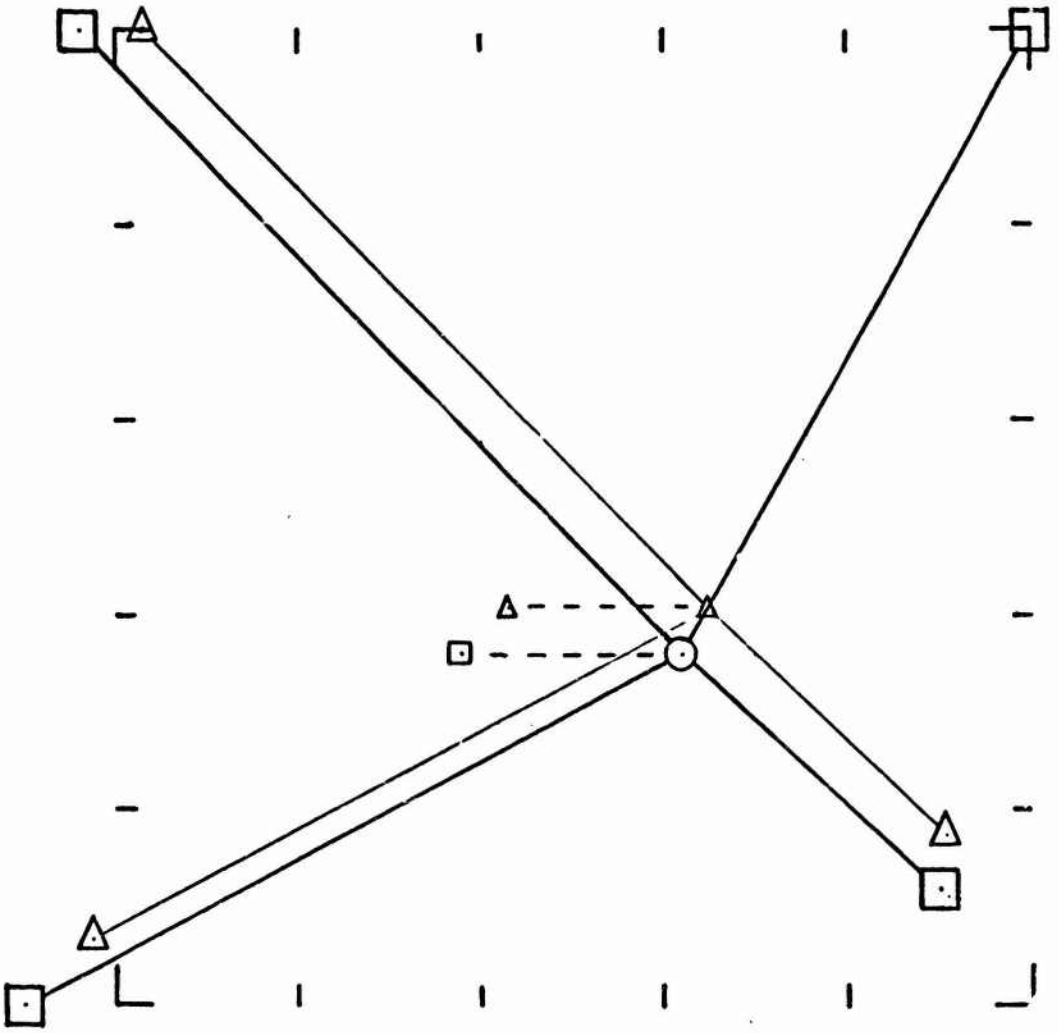


⊠ Target & Sensor Position

Baseline of Square Arrays

O 510	Φ 480
O 500	□ 470
D 490	Δ 460

Figure VIII, Cancellation of Random Bias Effects



Sensor Location

Calibrated Sensor Locations, 460M Array

Shell Impact Point

Target Position

Computed Position Target and Shell

COMPARISON OF POSITION TRACKING ACCURACY BY TARGET PATH ESTIMATION METHOD

TABLE IV

	POSITION TRACKING ERRORS					
	2 CONDITION		ALTERNATE 2 CONDITION		6 CONDITION	
	MEAN	STD DEV	MEAN	STD DEV	MEAN	STD DEV
RMS SENSOR ERROR (METERS)						
20.0	3.9	1.2	3.8	1.5	9.8	8.3
25.5	0.9	0.5	1.8	0.7	12.8	4.9
30.8	3.6	1.4	4.5	1.7	17.1	6.6
36.1	0.3	0.4	0.4	0.4	18.3	7.1
44.0	1.9	0.9	1.4	0.6	18.6	15.9
MISC. CASES						
1° STD DEV	15.3	4.6	19.7	5.8	20.6	19.7
6 DATA POINTS	10.3	2.22	9.5	2.1	25.2	5.6
8 DATA POINTS	3.6	1.7	3.5	1.8	12.8	3.8
10 DATA POINTS	3.6	2.5	2.1	2.0	72.2	27.6

used here, except that the bearings at a single target point are adjusted by an iterative least squares until the intercepts are coincident. Its use here would entail a two stage non-linear regression: first an iterative regression about each point along the target path to obtain the "best"estimate of position, followed by an iteration of the resection calibration regression to adjust sensor position, whereby the whole process would be repeated.

The number of bearings required to obtain a reasonable regression was evaluated for the 500 meter base case using a number of smoothed bearings from 10 to 2, Table V. Although the six point case did not suffer appreciable degradation, it is doubtful that this number would be sufficient to support field operations, especially in view of the need to perform pre-processing to obtain simultaneous bearings. If moving arc polynomials are used, a seven point arc would probably be required as a minimum, which increases the size of the set by six to a total of 12. Considerable reduction in the random noise present in the bearings can be expected from the inherent smoothing characteristic of this technique, and would partially offset the burden of additional data.

The effect of random noise in the angular data was evaluated for various one sigma values up to 3.0 degrees. A 494 meter array with relatively small random bias and a sensor placement error of 51.43 meters RMS was used. The set of bearings from the base case was modified by the appropriate one sigma value. A set of fixed random numbers were used to avoid large scale replications. Results are presented in Table VI and Figure IX. Based on this limited test, the effect on tracking accruacy behave in a relatively linear manner increasing the position error from approximately 1.5 meters to 15 meters for a one sigma value of 3 degrees. Applying the "Triangulation by Intersection" method described above, the effect of the bearing random noise is reduced to approximately 6 meters position error for the 3 degree one sigma case. The linear behavior however is no longer observed.

3. SUMMARY. Based on a limited evaluation of a linear target path through a 500 meter array, it appears that this calibration procedure can substantially improve the relative accuracy of an acoustic bearing sensor array. Mean position tracking errors of less than one percent of the array's basic size were obtained for sensor placement errors up to 50 meters RMS. Target velocity and azimuth headings to an equal or better accuracy were also obtained. This appears adequate to support target acquisition requirements for moving targets. It is expected that this performance could be improved by combining data from different target passes and including the impact points of ordnance items aimed at the targets.

Parametric analysis indicated that the procedure is sensitive to random noise in the angular data and random bias in sensor delivery errors. A somewhat shorter sample time than the 10 seconds used in this analysis may be required for numerical interpolation techniques.

TABLE V

REGRESSION ACCURACY
vs.
NUMBER OF BEARINGS
500 METER ARRAY
AND 47.4 METER RMS SENSOR ERROR

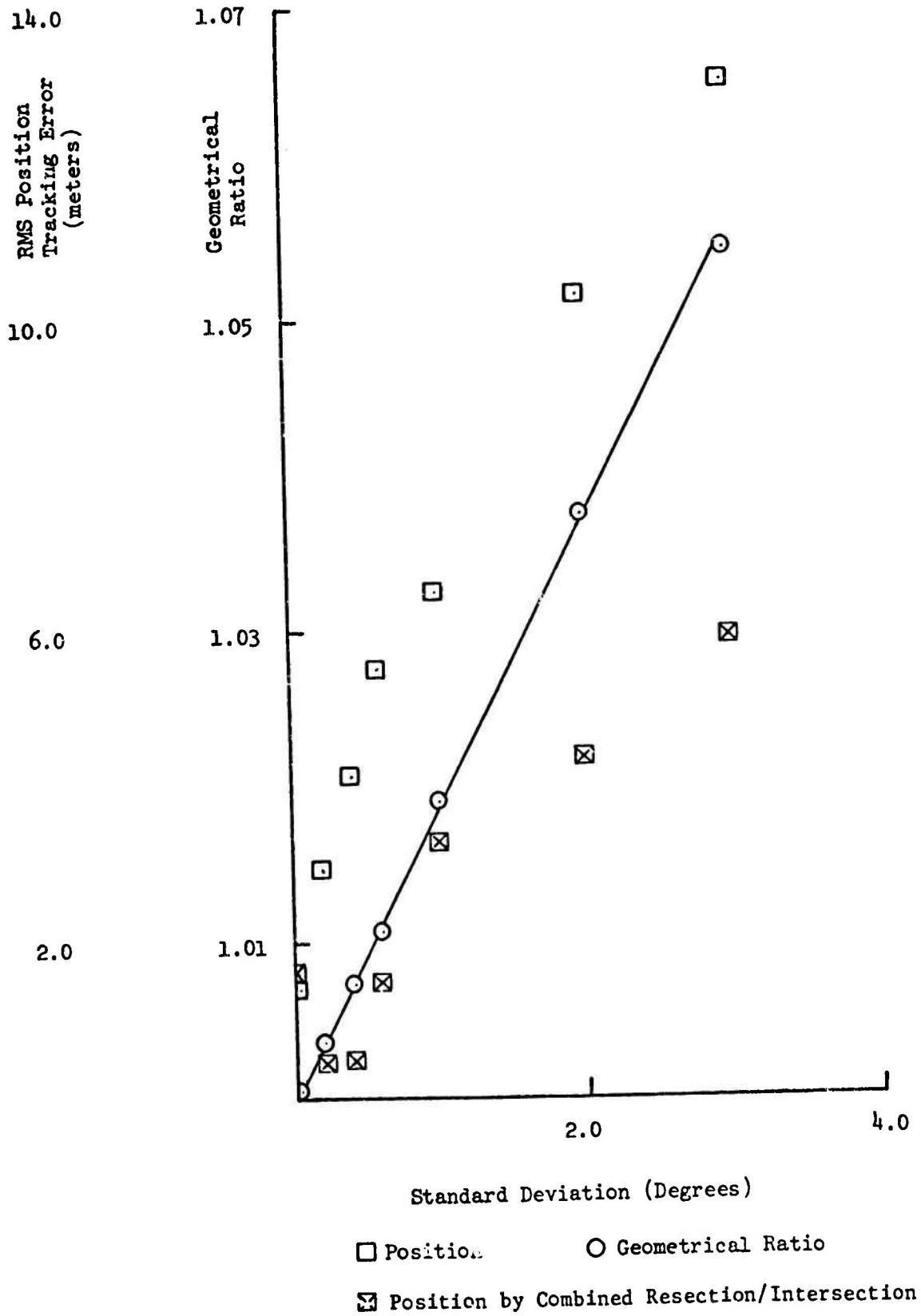
NUMBER OF SMOOTHED BEARINGS	REGR. RESID (M)	SENSOR POSITION ERROR		TARGET TRACKING ERRORS					
				POSITION (M)		VELOCITY (M/SEC)		AZIMUTH (DEG)	
	STD DEV	RMS (M)	GEO. RATIO	MEAN	STD DEV	MEAN	STD DEV	MEAN	STD DEV
10	2.02	6.42	.9852	3.62	2.47	.07	.13	−.54	.79
8	1.92	4.38	.9882	3.59	1.70	.07	.14	−.67	.76
6	1.96	8.37	1.002	10.26	2.22	.12	.15	−.47	.86
4	1.90	14.16	1.053	13.54	2.49	.19	.19	.45	1.15
2	.3E-3	49.89	1.27	19.53	5.04	.71	—	3.50	—

TABLE VI

REGRESSION SENSITIVITY TO BEARING RANDOM NOISE 494 METER ARRAY AND 51.43 METER RMS SENSOR ERROR

BEARING STANDARD DEVIATION (DEGREES)	REGR. RESID (M)	SENSOR POSITION ERROR		TARGET TRACKING ERRORS					
				POSITION (M)		VELOCITY (M/SEC)		AZIMUTH (DEG)	
		RMS (M)	GEO RATIO	MEAN	STD DEV	MEAN	STD DEV	MEAN	STD DEV
RESECTION									
0.01	0.497	1.36	1.0005	1.39	0.66	.022	.052	.015	0.130
0.2	1.720	2.12	1.0037	2.94	1.28	.034	.151	– .154	0.859
0.4	3.430	2.99	1.0073	4.15	2.04	.038	.296	– .185	1.623
0.6	5.097	3.95	1.0109	5.54	2.87	.046	.444	– .297	2.419
1.0	8.297	5.83	1.0190	6.53	4.52	.053	.734	– .474	4.019
2.0	15.811	10.61	1.0375	10.39	8.80	.080	1.480	–1.144	7.902
3.0	22.761	15.74	1.0546	13.62	13.24	.143	2.211	–1.898	11.721
COMBINED RESECTION/ INTERSECTION									
0.01	0.49	1.23	1.0005	1.59	0.66	.015	.052	.018	.132
0.2	1.71	1.53	1.0039	0.44	0.95	.012	.144	– .109	.737
0.4	3.41	2.40	1.0078	0.47	1.67	.007	.281	– .085	1.363
0.6	5.07	3.36	1.0115	1.47	2.46	.012	.419	– .151	2.029
1.0	8.26	5.42	1.0184	3.29	4.01	.026	0.681	–0.287	3.333
2.0	15.73	11.26	1.0401	4.39	8.27	.043	1.385	–0.564	6.850
3.0	22.63	17.10	1.0583	5.93	12.72	.106	2.063	–1.17	10.86

Figure IX, Position Error and Geometrical Ratio vs. Bearing Random Noise



APPENDIX A - MODEL DERIVATION

The coordinates of the intercept points can be determined by the simultaneous solution of 2 linear point-slope equations.

$$X_{121} = C_{121}\,[(Y_2 - Y_1) + (M_{11}\,X_1 - M_{21}\,X_2)] \qquad (1)$$

$$Y_{121} = C_{121}\,[(M_{11}Y_2 - M_{21}Y_1) + M_{11}M_{21}\,(X_1 - X_2)]$$

$$\vdots$$

$$X_{341} = C_{341}\,[(Y_4 - Y_3) + (M_{31}X_3 - M_{41}\,X_4)]$$

$$Y_{341} = C_{341}\,[(M_{31}Y_4 - M_{41}Y_3) + M_{31}M_{41}(X_3 - X_4)]$$

$$\vdots$$

$$X_{ijk} = C_{ijk}\,[(Y_j - Y_i) + (M_{ik}X_i - M_{jk}X_j)]$$

$$Y_{ijk} = C_{ijk}\,[(M_{ik}Y_j - M_{jk}Y_i) + M_{ik}M_{jk}\,(X_i - X_j)]$$

for

$$i, j = 1, 2, 3, 4 \text{ and } i \neq j$$

where

X_{ijk}, Y_{ijk} - predicted target location at k^{th} point. by sensors i and j

X_i, Y_i - assumed position i^{th} sensor

M_{ik} - tangent of bearing measured by sensor i in the k set

$C_{ijk} = 1/(M_i - M_j)$.

A set of 12 equations exists for every point along the path for which simultaneous bearings can be obtained. These equations consist of six pairs in X and Y corresponding to each side and diagonal of a quadrangle. Since any pair is sufficient to calculate target position, the set is over-determined, and the X_{ij}, Y_{ij} points will vary unless the

data are error free.

Expanding Equations 1 about the initial sensor locations by Taylor's series, and dropping higher order terms,

$$X_k = F_{xijk} + a_i \Delta X_i + a_j \Delta X_j + a_{i+4} \Delta Y_i + a_{j+4} \Delta Y_j;$$
$$Y_k = F_{yijk} + a'_i \Delta X_i + a'_j \Delta X_j + a'_{i+4} \Delta Y_i + a'_{j+4} \Delta Y_j; \qquad (2)$$

for

$$i, j = 1, 2, 3, 4;\ i \neq j;\ \text{and}\ \Delta X_1 = \Delta Y_1 = 0$$

where

ΔX_i, ΔY_i are sensor position corrections

F_x and F_y are given by Equations 1 at X_o and Y_o

$$a_i = \frac{\partial F_{xij}}{\partial X_i} = C_{ij} M_i \qquad a_{i+4} = \frac{\partial F_{xij}}{\partial Y_i} = C_{ij}$$

$$a'_i = \frac{\partial F_{yij}}{\partial X_i} = C_{ij} M_i M_j \qquad a'_{i+4} = \frac{\partial F_{yij}}{\partial Y_i} = C_{ij} M_j$$

a set of linear equations is obtained for each target position as a function of sensor position corrections. The partial derivates are interpreted as the GEDOP numbers relating triangulation errors to baseline variations. For this to be completely valid, one of the sensor's position must be constant such that the variations in x and y correspond to a change in the magnitude of the baseline rather than a translation. Otherwise, the partials must be taken with respect to the baselines rather than the x, y sensor coordinates. Also, since three vectors with common origin are sufficient to describe a quadrangle, only three sensor locations relative to the fourth can be determined.

Expressing the target position given by Equations 2 in terms of an estimated value, such as the arithmetic average of two sensor pairs, and a residual error term,

$$X_k = \bar{X}_k - R_{xijk} \qquad Y_k = \bar{Y}_k - R_{yijk} \qquad (3)$$

the total set of 12k intercept equations can be written as

$$R + AS + F = 0 \qquad (4)$$

where

$$\bar{R} = \begin{bmatrix} R_{x121} \\ R_{y121} \\ \vdots \\ R_{y34k} \end{bmatrix} \qquad A = \begin{bmatrix} \frac{\partial F_{x121}}{\partial X_1} & \frac{\partial F_{x121}}{\partial X_2} & \cdots & \frac{\partial F_{x121}}{\partial Y_3} \\ \frac{\partial F_{y121}}{\partial X_1} & \frac{\partial F_{y121}}{\partial X_2} & \cdots & \frac{\partial F_{y121}}{\partial Y_3} \\ \vdots & & & \\ \frac{\partial F_{y34k}}{\partial X_1} & \frac{\partial F_{y34k}}{\partial X_2} & \cdots & \frac{\partial F_{y34k}}{\partial Y_3} \end{bmatrix}$$

$$S = \begin{bmatrix} \Delta X_1 \\ \Delta X_2 \\ \Delta X_3 \\ \Delta Y_1 \\ \Delta Y_2 \\ \Delta Y_3 \end{bmatrix} \qquad F = \begin{bmatrix} F_{x121} - \bar{X}_1 \\ F_{y121} - \bar{Y}_1 \\ \vdots \\ F_{y341} - \bar{Y}_1 \\ \vdots \\ F_{y34k} - \bar{Y}_k \end{bmatrix}$$

The normal equation for least squares adjustment is

$$S + (A^tA)^{-1}A^tF = 0. \qquad (5)$$

The predicted target locations can now be forced to converge, by solving the normal equation for the solution vector S, adjusting the sensor positions by $\Delta X_i, \Delta Y_i$ (Components of the S vector). recomputing F which now includes a revised target estimate, and repeating the process until the correction components of S go to zero or become sufficiently small. The convergence achieved by the regression is expressed in terms of the standard deviation of the residuals which can be determined by solving equation 4, R = -F.

The effect of random noise in the bearing data can also be reduced by minimizing the sum of the squared intercept deviations about some point. Expanding the intercept equations given in 1 about the slope by Taylor's series and dropping higher order terms,

$$X_k = F_{xijk} + b_i\Delta\theta_{ik} + b_j\Delta\theta_{jk} \qquad (6)$$

$$Y_k = F_{yijk} + B_i\Delta\theta_{ik} + B_j\Delta\theta_{jk}$$

for

$i, j = 1, 2, 3, 4$, and $i \neq j$,

where

F_{xijk} and F_{yijk} are given by Equation 1 at X_o, Y_o, θ_o

$\Delta\theta_i, \Delta\theta_j$ are bearing corrections

$$b_i = \frac{\partial F_{xijk}}{\partial\theta_i} = C_{ijk}(X_i - F_{xijk})(1 + M_j^2)$$

$$b_j = \frac{\partial F_{xijk}}{\partial \theta_j} = C_{ijk} (X_j - F_{xijk}) (1 + M_j^2)$$

$$b_i = \frac{\partial F_{yijk}}{\partial \theta_i} = C_{ijk} [Y_j + M_j (X_i - X_j) - F_{yijk}] (1 + M_i^2)$$

$$b_j = \frac{\partial F_{yijk}}{\partial \theta_j} = C_{ijk} [Y_i + M_i (X_j - X_i) - F_{yijk}] (1 + M_j^2)$$

a set of linear equations is obtained for each of the fours sensor locations in terms of bearing corrections.

Utilizing the estimated target position given in Equation 3, the set of eight intersection equations can be written in the form of Equation 4

$$R + B\Delta\theta + F = 0. \qquad (7)$$

Solving in the manner previously used for Equation 5, the X_k, Y_k values determined by intersecti can now be substituted for the estimated target values, Equation 3, used in the resection regression.

ATTENDEES

1974 ARMY NUMERICAL ANALYSIS CONFERENCE
13 and 14 February 1974

U. S. Army Frankford Arsenal
Philadelphia, Pennsylvania 19137

Charles N. Alston
USA Materiel Sys Anal Agency
Aberdeen Proving Ground, Md. 21005

Dr. Taylan Altan
Battelle Columbus Laboratories
Columbus, Ohio 43201

Larry Ihrig Amstutz
USAMERDC
Fort Belvoir, Virginia 22060

R. W. Anderson, Jr.
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Dr. John C. Atkinson
Food and Drug Administration
Washington, D.C. 20204

Leroy Baer
Edgewood Arsenal
Aberdeen Proving Ground, Md. 21005

Richard J. Bair
Watervliet Arsenal
Watervliet, New York 12189

Robert Barnas
Picatinny Arsenal
Dover, New Jersey 07801

Bruce Barnett
Picatinny Arsenal
Dover, New Jersey 07801

CPT A. G. Bonifas
US Military Academy
West Point, New York 10996

William Brankowitz
Edgewood Arsenal
Aberdeen Proving Ground, Md. 21005

J. Thomas Broach
USAMERDC
Fort Belvoir, Virginia 22060

David Brown
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Louis R. Cerrato
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Jagdish Chandra
US Army Research Office
Durham, North Carolina 27706

Raymond F. Coakley, Jr.
USAMERDC
Fort Belvoir, Virginia 22060

Douglas G. Conley
USAMERDC
Fort Belvoir, Virginia 22060

Fred Crary
University of Wisconsin-MRC
Madison, Wisconsin 53706

Eleanor R. Cross
Walter Reed Army Med Cen
Washington, D.C. 20012

Elizabeth H. Cuthill
Naval Ship Res and Dev Cen
Bethesda, Maryland 20034

The last page of this attendence list contains some names not in alphabetical order.

Attendees -- 1974 Army Numerical Analysis Conference

Diana Dadamo
Frankford Arsenal
Philadelphia, Pennsylvania 19137

James R. Davis
ACCH-DS-PT
Fort Huachuca, Arizona 85613

Lynn Davis
Edgewood Arsenal
Aberdeen Proving Ground, Md. 21005

Kenneth J. Dean
USAMERDC
Fort Belvoir, Virginia 22060

C. de Boor
University of Wisconsin-MRC
Madison, Wisconsin 53706

Saranne Dix
USA Mgmt Sys Support Agency
Washington. D.C. 20310

Phillip L. Doiron, Sr.
Waterways Experiment Station
Vicksburg, Mississippi 39180

Dr. Louis D. Duncan
US Army Electronics Command
White Sands Missile Range, NM 88002

Russell Eaton, III
USAMERDC
Fort Belvoir, Virginia 22060

A. Gerald Edwards
Picatinny Arsenal
Dover, New Jersey 07801

Sylvan Eisman
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Dr. M. Zaki El-Sabban
USA Aviation Systems Command
St. Louis, Missouri 63166

Dr. Walter D. Foster
Army Surgeon General, HQDA-DASG-ISP
Washington, D.C. 20314

Clinton M. Frank
Ballistics Research Laboratories
Aberdeen Proving Ground, Md. 21005

Diana Frederick
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Lawrence A. Gambino
USA Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060

Edward H. Gamble
USA Test and Evaluation Command
Aberdeen Proving Ground, Md. 21005

John Giese
Ballistic Research Laboratories
Aberdeen Proving Ground, Md. 21005

Paul Gordon
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Dallas Deryl Haddox
Defense Mapping Agency
Washington, D,C.

Harold P. Hammann
USA Troop Support Command
St. Louis, Missouri 63120

J. L. Harris
USA Missile RD&E Laboratory
Redstone Arsenal, Alabama 35809

Ralph Harris
USA Mgmt Engr Trng Agency
Rock Island, Illinois 61201

Robert H. Haveson
Picatinny Arsenal
Dover, New Jersey 07801

Attendees -- 1974 Army Numerical Analysis Conference

Magnus R. Hestenes
UCLA and IBM Research
Yorktown Heights, New York 10598

Dr. Edmund H. Inselmann
USA Materiel Command
Alexandria, Virginia 22304

Joseph M. Iseman
Harry Diamond Laboratories
Washington, D.C.

Robert Isakower
Picatinny Arsenal
Dover, New Jersey 07801

Henry Kahn
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Orrin C. Kaste
Ballistic Research Laboratories
Aberdeen Proving Ground, Md. 21005

Joseph W. Kaszupski
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Dr. Chul Kim
USA Natick Laboratories
Natick, Massachusetts 01760

Frederick G. King
USA Materiel Sys Anal Agency
Aberdeen Proving Ground, Md. 21005

Leon E. Klarman
USA Natick Laboratories
Natick, Massachusetts 01760

June Kryst
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Robert LaBudde
University of Wisconsin-MRC
Madison, Wisconsin 53706

Jerome E. Lattery
USA Corps of Engineers
Livermore, California

Eric L. Leese
National Defense Headquarters
Ottawa, Ontario, Canada

Melanie L. Lenard
University of Wisconsin-MRC
Madison, Wisconsin 53706

Leon Leskowitz
USA Electronics Command
Fort Monmouth, New Jersey 07703

John F. McNamara
USA Const Engr Res Lab
Champaign, Illinois 61880

Joseph B. Marburger
USA Mobility Equipment R&D Center
Fort Belvoir, Virginia 22060

Richard T. Maruyama
USA Materiel Sys Anal Agency
Aberdeen Proving Ground, Md. 21005

Dr. Ceslovas Masaitis
Ballistic Research Laboratory
Aberdeen Proving Ground, Md. 21005

John M. Meredith
USA Materiel Sys Anal Agency
Aberdeen Proving Ground, Md. 21005

Colonel Lothrop Mittenthal
USA Research Office
Durham, North Carolina 27706

Howard Moskowitz
USA Natick Laboratories
Natick, Massachusetts 01760

Dr. Theodosis Pavilides
Princeton University
Princeton, New Jersey

Attendees -- 1974 Army Numerical Analysis Conference

Harry X. Peaker
USA Materiel Sys Anal Agency
Aberdeen Proving Ground, Md. 21005

Edward K. Pedersen
USA Mgmt Sys Support Agency
Washington, D.C. 20310

Newton Penrose
USA Military Academy
West Point, New York

Kenneth Pfleger
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Dr. Mel W. Pirtle
NASA, Ames Research Center
Moffett Field, California 94035

Prof. Louis B. Rall
University of Wisconsin-MRC
Madison, Wisconsin 53706

Fredrick L. Roder
USAMERDC
Fort Belvoir, Virginia 22060

J. Barkley Rosser
University of Wisconsin-MRC
Madison, Wisconsin 53706

Dr. William A. Sacco
Edgewood Arsenal
Aberdeen Proving Ground, Md. 21005

Aldric Saucier
USA Materiel Command
Alexandria, Virginia 22304

Ferdinand Scerbo
Picatinny Arsenal
Dover, New Jersey 07801

Wright H. Scidmore
Frankford Arsenal
Philadelphia, Pennsylvania 19137

John Seigh
Edgewood Arsenal
Aberdeen Proving Ground, Md. 21005

Dwight M. Shaw
USA Mobility Equipment R&D Center
Fort Belvoir, Virginia 22060

James W. Shean
Frankford Arsenal
Philadelphia, Pennsylvania 19137

William Shulman
Edgewood Arsenal
Aberdeen Proving Ground, Md. 21005

I. H. Slook
Frankford Arsenal
Philadelphia, Pennsylvania 19137

George R. Staton
Frankford Arsenal
Philadelphia, Pennsylvania 19137

David Steinberg
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Alvin K. Takemoto
USA Mgmt Engr Trng Agency
Rock Island, Illinois 61201

Dr. Ronald P. Uhlig
USA Materiel Command
Alexandria, Virginia 22304

Allen Weinberger
USAMERDC
Fort Belvoir, Virginia 22060

Lt. Col. William Whittaker
DDR&E
Washington, D.C.

Bruce Wilson
Dugway Proving Ground
Dugway, Utah 84022

Attendees -- 1974 Army Numerical Analysis Conference

Fred Witt
Frankford Arsenal
Philadelphia, Pennsylvania 19137

James T. Wong
Air Mobility R&D Laboratory
Moffett Field, California 94035

James Wood
Edgewood Arsenal
Aberdeen Proving Ground, Md. 21005

Gary W. Woods
Watervliet Arsenal
Watervliet, New York 12189

W. T. Abel
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Arthur D. Grainger
Defense Mapping Agency Topographic Ctr
6500 Brooks Lane
Washington, D.C.

Luis Marquez
TE-AS Weapon Simulation Branch
White Sands Missile Range, NM 88002

Dr. Stephen M. Robinson
Mathematics Research Center
University of Wisconsin
Madison, Wisconsin 53706

Alfonso G. Wright
Rock Island Arsenal
Rock Island, Illinois 61201

Dr. Julian J. Wu
Watervliet Arsenal
Watervliet, New York 12189

J. M. Yohe
University of Wisconsin-MRC
Madison, Wisconsin 53706

Elena Zagustin
Calif. State University Long Beach
Long Beach, California 90840

Tobey J. Reed
USA Natick Laboratories
Natick, Massachusetts 01760

Gordon Sigman
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Byron White
Picatinny Arsenal
Dover, New Jersey 07801