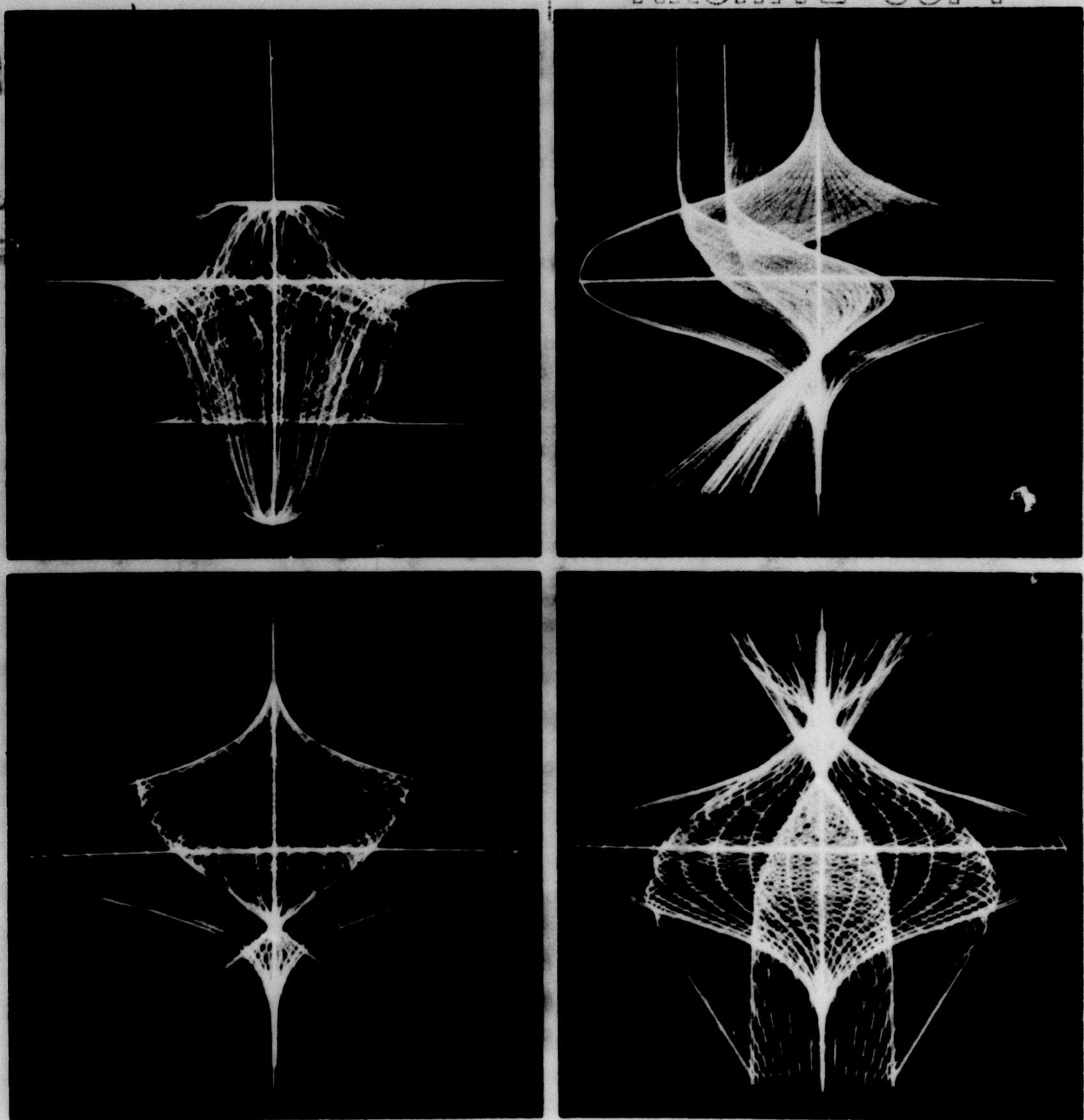
Massachusetts Institute of Technology

Project MAC Progress Report II July 1964 to July 1965

Code 1



X

Massachusetts Institute of Technology
Project MAC
545 Technology Square
Cambridge, Massachusetts
02139

Work reported herein was supported by Project MAC, an M. I. T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(01). The primary support for some of this work came from the M. I. T. departments and laboratories participating in Project MAC, whose research programs are, in turn, sponsored by various government and private agencies. This support is acknowledged by specific mention of agency and contract number in the appropriate sections.

Government contractors may obtain copies of this Progress Report, and all Project MAC Technical Reports listed in Appendix D, from the Defense Documentation Center, Document Service Center, Cameron Station, Alexandria, Virginia 22314. Orders will be expedited from DDC if placed by your librarian, or some other person authorized to request documents.

Other U. S. citizens and organizations may obtain copies of the above items from the Clearinghouse for Federal Scientific and Technical Information (CFSTI), Sills Building, 5285 Port Royal Road, Springfield, Virginia 22151.

BLANK PAGE

ARTIFICIAL INTELLIGENCE

BIOLOGY DEPARTMENT

CIVIL ENGINEERING DEPARTMENT

COMPUTER COMMUNICATION STRUCTURES

COMPUTER OPERATION

COMPUTER SYSTEM RESEARCH

ELECTRONIC SYSTEMS LABORATORY

LIBRARY RESEARCH

LINCOLN LABORATORY

NON-M.I.T. USERS

OPERATIONS RESEARCH CENTER

RESEARCH LABORATORY OF ELECTRONICS

SCHOOL OF ENGINEERING

SCHOOL OF HUMANITIES AND SOCIAL SCIENCES

SCHOOL OF SCIENCE

SLOAN SCHOOL OF MANAGEMENT

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

PERSONNEL
TO JULY 1, 1965

Administration

Prof. R. M. Fano
- Director

O. G. Selfridge
- Associate Director

R. G. Mills
- Assistant Director

Committee On Information Processing

Prof. P. M. Morse, Chairman
Dean G. S. Brown
Prof. P. Elias
Prof. R. M. Fano
Prof. R. I. Hulsizer
Dean H. W. Johnson
Prof. C. L. Miller
Prof. C. F. J. Overhage
Dean J. B. Wiesner

Academic Staff

S. S. Alexander
A. E. Amstutz
G. W. Angell
A. Bers
J. M. Biggs
M. Blum
D. G. Bobrow
J. F. Brotchie
D. C. Carroll
S. A. Coons
F. J. Corbató
J. B. Dennis
M. L. Dertouzos
A. G. H. Dietz
W. M. Evan
R. M. Fano
D. E. Farrar
W. R. Fey
J. W. Forrester
W. D. Getty
M. Greenberger
M. Halle
K. F. Hansen
H. A. Haus
J. M. Heinz
L. N. Howard
R. A. Howard
D. A. Huffman
R. Y. Kain
R. Kaplow
D. S. Kuck
E. Kuh
T. W. Lambe
C. Levinthal
J. D. C. Little
C. L. Liu
R. D. Logcher
R. McNaughton
C. L. Miller
M. L. Minsky
P. M. Morse
C. F. J. Overhage
J. W. Poduska
I. D Pool
J. F. Reintjes
E. B. Roberts
J. M. Roesset
A. L. Samuel
R. L. Schiffman
C. E. Shannon
L. D. Smullin
T. G. Stockham, Jr.
G. M. Sturman
H. M. Teager
D. E. Troxel
J. R. Walton
H. M. Weingartner
J. Weizenbaum
R. V. Whitman

Research Associates, Instructors and Lecturers

J. W. Brackett
E. P. C. Fernando
E. L. Glaser
G. A. Gorry, Jr.
A. M. Hershdorfer
M. M. Jones
R. Langridge
D. N. Ness
S. Papert
A. L. Pugh, III
D. Roos
J. H. Saltzer
H. M. Schneider
J. Sussman
M. E. Wantman
J. W. Weber
S. H. Whitelaw
V. H. Yngve

(Non-Academic) Research Staff

A. W. Armenti
R. F. Auzas
M. J. Bailey
E. C. Bartels
R. Bayles
F. Belvin
G. L. Benton
J. W. Bex
R. J. Bigelow
G. M. Bonham
C. M. Bosche
M. L. Cabral
E. Campbell
J. Casey
M. E. Child
G. Clancy
A. Congleton
R. Creasy
P. A. Crisman
M. C. Crocker, III
C. B. Cushing
R. C. Daley
J. L. Darlington
J. M. Dolan
E. J. Dole
S. D. Dunten
R. S. Fabry
C. G. Feldmann
R. Fenichel
A. T. Foster
G. Fratar
C. Garman
R. V. Goodman
R. M. Graham
U. F. Gronemann
J. A. Gunn
D. R. Haring
T. Hastings
M. C. Henneman
F. B. Hills
J. M. Hodnick
R. Hussey
B. Johanson
D. Johnson
I. Johnson
M. S. Kannel
J. Katzenelson
M. M. Kessler
S. Kidd
E. M. Kliman
H. K. Knudsen
P. T. Ladd
R. O. Ladson
C. A. Lang
R. B. Lapin
R. Leopold
T. Lerner
M. Levin
D. Levine
E. R. Lewis
J. W. Lucey
G. A. Maley
W. D. Mathews
W. D. Maurer
S. McIntosh
V. E. McLoud
E. Miller
R. G. Mills
N. I. Morris
G. R. Murray
J. Nagle
W. R. Noftsker
J. F. Nolan
R. Orenstein
J. A. Parisot
M. M. Pennell
H. C. Peterson
T. Pitidis-Poutous
R. B. Polansky
L. Pouzin
K. Reinschmidt
E. River
D. T. Ross
A. Rutchka

Non-Academic Research Staff (continued)

S. Sacks
A. J. Saltalamacchia
P. Samson
R. A. Sayers
G. C. Schroeder
P. M. Sheehan
M. V. Solomita
R. L. South
J. E. Spencer
N. C. Spencer
R. H. Stotz
K. A. Szabo
D. E. Thornhill
T. H. Van Vleck
L. Varian
M. R. Wagner
J. F. Walsh
R. A. Walter
J. E. Ward
D. B. Yntema
R. J. Zatorski

Resident Guests

C. W. Bower - Boeing Company
R. K. Coe - United Aircraft Corp.
B. T. Fox - Sandia Corp.
L. H. Haines - I. B. M. Corp.
J. R. Kennedy - Lockheed-Georgia Co.
J. Martyniak - North American Aviation, Inc.
D. E. Oppert - National Security Agency
I. C. Pyle - AERE, Harwell, England
H. W. Spencer - Grumman Aircraft Engr'g Co.

Non-M. I. T. Users

D. Drukey	Systems Development Corporation
M. M. Flood	University of Michigan
A. Leon	University of Michigan
S. Lorch	Massachusetts General Hospital
J. McCarthy	Stanford University
M. T. McGuire	Massachusetts General Hospital
A. Oettinger	Harvard University
S. M. Pizer	Harvard University
G. C. Quarton	Massachusetts General Hospital
A. Ruyle	Harvard University
J. E. K. Smith	University of Michigan
P. J. Stone	Harvard University
L. Uhr	University of Michigan
K. Winiecki	Harvard University

Research Assistants

T. H. Asselin
W. Bailey
L. N. Beckreck
D. Beltran-Maldonado
J. R. Brach
R. H. Campbell
T. B. Cheek
P. Clermont
D. J. Edwards
R. E. Efimba
D. S. Evans
G. C. Everest
L. Fattal
H. C. Frazier, Jr.
R. H. González-Zubieta
A. D. Griffith
G. Guzman Baron-Torres
E. G. Guttmann
T. P. Hart
N. B. Heubeck
J. Huffman
E. L. Ivie
T. J. Johnson
L. S. Kanodia
A. Kessler
B. V. Koen
J. F. Kramer
J. Ku
B. R. Kusse
M. M. Lind
W. H. Linder
J. L. Linderman
G. C. Ling
D. T. Llewellyn-Jones
A. W. MacEwan
A. Malhotra
W. A. Martin
S. G. Mazzotta
L. F. McPherson, III
J. R. Miller, III
R. L. Miller
J. D. Mills
J. Moses
O. C. Nord
L. M. Norton
P. C. Ordeshook
R. Parmelee
A. Pawlikowski
W. M. Pecknold
G. Piotrowski
J. E. Rodriguez
R. C. Rosenberg
J. R. Roy
D. D. Ryu
P. J. Santos, Jr.
E. H. Sibley
C. E. Speck
W. D. Stratton
S. L. Strong
L. C. Teague
W. Teitelman
C. C. Tillman, Jr.
E. C. Van Horn, Jr.
M. Weinberger
J. I. Weiner
P. A. Wieselmann
D. U. Wilde
B. L. Wolman

Other Graduate Students

B. H. Bloom
G. J. Burnett
A. A. Bushkin
G. D. Y. Chang
W. Y. Chang
P. B. Clark
R. Crossley
J. E. Daily
J. A. Davis
P. Denning
T. J. Fessenden
Z. Fluhr
R. C. Gammill
R. A. Hadley
M. Hamilton
H. F. Ledgard
F. F. Lee
M. A. Lieberman
F. L. Luconi
M. Manove
Q. L. Miller
J. H. Morris, Jr.
D. B. Murton
K. Olshansky
S. Popkin
A. S. Priver
E. A. Robertson
F. J. Russo
H. S. Schwenk, Jr.
C. L. Seitz

Other Graduate Students (continued)

H. L. Selesnick	S. Spitzer	A. D. Weiss
W. Selles	J. L. Sussman	E. Westerfeld
L. L. Selwyn	C. W. Therrien	J. M. Winett
A. A. Smith	B. T'sou	S. Zucker
	R. N. Wallace	M. Zwick

Undergraduates

W. C. Albertson	G. Grover	A. Reed
A. E. Armstrong	R. V. Harris	G. L. Rosen
P. Bos	R. Hodges	K. Rosenfeld
A. Buckles	T. R. Hofmann	L. J. Rotenberg
R. L. Daniels	H. Ishii	R. Sayre
D. S. Diamond	M. E. Kaliski	D. H. Slosberg
A. A. Dvorak, Jr.	E. Kanstroom	G. J. Sussman
H. Ellis	A. N. Kramer	B. J. Vander Molen
C. Ellison	D. A. Lasher	S. A. Ward
J. Ever	H. Levin	D. R. Widrig, Jr.
P. Greene	P. C. Lynch	P. M. Wolk
D. Griffel		

Technical Assistants

G. Blum	J. Rizika	A. W. Wilson
J. T. Holloway	C. H. Willson	

Technicians

R. Fidler	R. J. Porcaro

Operating Staff

G. Andrews
D. Borsini
R. Brenner
K. Carley
R. Cerrato
G. Dardis
T. Dattilo
R. A. Degan
R. G. Hart
R. McNamara
R. Moore
S. Nelson
G. Noseworthy
M. Pagliarulo
R. Parker
E. Reardon
J. W. Waclawski

Administrative and Supporting Staff

F. S. Axelrod
M. E. Baker
A. W. Bowen
J. M. Constantine
F. H. Dilworth
E. T. Gannon
M. Grourke
C. L. King
L. Martin
S. A. O'Leary
J. E. Pinella
C. L. Robinson
C. Sarandrea
D. C. Scanlon
E. L. Schneider
L. M. Silvestro
J. Wallen
R. M. Yunetz

Project MAC was organized at the Massachusetts Institute of Technology in the spring of 1963 for the purpose of conducting a research and development program on Machine-Aided Cognition and Multiple-Access Computer systems. It operates under contract with the Office of Naval Research, acting on behalf of the Advanced Research Projects Agency of the Department of Defense.

The broad goal of Project MAC is the experimental investigation of new ways in which on-line use of computers can aid people in their individual intellectual work, whether research, engineering design, management, or education. One envisions an intimate collaboration between man and computer system in the form of a real-time dialogue where both parties contribute their best capabilities. Thus, an essential part of the research effort is the evolutionary development of a large, multiple-access computer system that is easily and independently accessible to a large number of people, and truly responsive to their individual needs. The MAC computer system is a first step in this direction and is the result of research initiated several years ago at the MIT Computation Center.

Project MAC was organized in the form of an interdepartmental, inter-laboratory "project" to encourage widespread participation from the MIT community. Such widespread participation is essential to the broad, long-term project goals for three main reasons: exploring the usefulness of on-line use of computers in a variety of fields, providing a realistic community of users for evaluating the operation of the MAC computer system, and encouraging the development of new programming and other computer techniques in an effort to meet specific needs.

Faculty, research staff, and students from fourteen academic departments and four interdepartmental research laboratories are participating in Project MAC. For reporting purposes, they are divided into sixteen groups, whose names correspond in many cases to those of MIT schools, departments and research laboratories. Some of the groups deal with research topics that fall under the heading of computer sciences; others with research topics which, while contributing in a substantive way to the goals of Project MAC, are primarily motivated by objectives outside the computer field.

The purpose of this Progress Report is to outline the broad spectrum of research being carried out as part of Project MAC. Internal memoranda of Project MAC are listed in Appendix A, and MAC-related theses are listed in Appendix B. Some of the research is cosponsored by other governmental and private agencies, and its results are described in journal articles and reports emanating from the various MIT departments and laboratories participating in Project MAC. Such publications are listed in Appendix C of the report. Project MAC Technical Reports are listed in Appendix D.

Robert M. Fano
Cambridge, Massachusetts

ARTIFICIAL INTELLIGENCE

A Visually-Controlled Manipulator

Unrecognizable Sets of Numbers

The Algebraic Approach to Finite Automata

The Design of LISP 2

Man-Computer Symbiosis

Mathematical Assistant

Heuristics of Theorem Proving in Group Theory

Computer Experiments in Finite Algebra

A Chess-Playing Program

MATHLAB: On-Line Symbolic Computation

INTEGRATE: On-Line Indefinite Integration

Academic Staff

M. L. Minsky
M. Blum
D. G. Bobrow
S. Papert
A. L. Samuel

Non-Academic Research Staff

G. Blum
D. J. Edwards
C. Engleman (MITRE Corp.)
J. T. Holloway
M. Levin
W. D. Maurer
W. R. Noftsker
P. Samson
R. L. South

Research Assistants and other Students

B. H. Bloom
A. D. Griffith
T. P. Hart
M. Manove (MITRE Corp.)
W. A. Martin
J. Moses
L. M. Norton
A. Reed
G. J. Sussman
W. Teitelman

A Visually-Controlled Manipulator - Marvin L. Minsky

Since the completion of a computer-controlled mechanical hand in 1961 (H. Ernst, Ph. D. Thesis, M.I.T., 1961) little has been done in the area of design of autonomous manipulators. We have been working the development of a versatile, computer-operated, visually-oriented machine for handling objects in complicated spatial situations. A real-time, on-line, television-camera interface has been built for the PDP-6 computer and has been used to track the motion of simple visual objects, including the human eye. This is of interest in studying the mechanism of human visual preception. Also, an AMF Versatran, Model C, mechanical remote manipulator has been coupled to the PDP-6.

Unrecognizable Sets of Numbers - Marvin L. Minsky and Seymour Papert

The problem here is, given a set A of non-negative integers, is there a finite-state device that will discriminate between members and non-members of A when the numbers are presented in binary, or some other similar number-representation system with a different radix (such as ternary of decimal)? A technique for showing that certain sets are not recognizable has been written. The technique deals with the asymptotic behavior of the function π_A (n), which is the number of integers less than n in the set A; unless π_A (n) has a specified behavior, A is not recognizable. However, not all sets A with a satisfactory asymptotic behavior of π_A (n) are recognizable. Another technique developed by McNaughton handles some of these cases. The goal for research in this area is to determine a necessary and sufficient condition that A be recognizable in r-ary ($r \geqq 2$) by a finite state device. A paper will appear soon in the Journal of the ACM.

The Algebraic Approach to Finite Automata - Seymour Papert and Robert NcNaughton

The work by Schutzenberger, Rhodes and Krohn, and some of our own results have convinced us of the fruitfulness of the algebraic approach to finite automata, and, in particular, the concept of the semi-group of the machine. The two problems that have been most impressively dealt with are the problem of decomposition of machines, and the problem of general star height. It is our intention to continue to focus on this area. Research, however, will lean toward application of already existing knowledge to problems about machines, rather than advancing knowledge in the field of algebra. We have been concerned with simplifying some of the proofs of known theorems (applying algebra to machines) both to make fundamental notions clear enough for future research and thinking, and also to bring these concepts within the purview and parlance of beginning students in the field.

It is felt that use of the computer will be fruitful in this area. Some of the programmers in the Artificial Intelligence Group have been writing LISP programs that will do such things as find the semigroup of a given finite-state machine. Use of a time-shared computer in this way can be very useful in checking out hunches in the search for valid generalities or in providing counter-examples.

The Design of LISP 2 - Daniel G. Bobrow

The design of a new programming language, called LISP 2, is now being carried out jointly by Systems Development Corporation and Information International Incorporated, with aid from members of the Project MAC Artificial Intelligence Group. LISP 2 will be an ALGOL-style language and will have the desirable ALGOL features of dynamic allocation of storage, recursive functions, and dynamic arrays. LISP 2 will have list processing as in the current LISP 1.5 language; however, the control language will be in ALGOL rather than in LISP. In this list processing, useful list structures will be compacted to provide room for the allocation of space for arrays and for new list-structure elements.

In addition to the LISP 1.5 function-oriented list processing, LISP 2 will embody format-directed string processing similar to that of the COMIT programming language. With this processing a user can specify a pattern against which a string is matched. If it is matched successfully, the string is parsed, that is, subdivided into segments which match the individual patterns in the given pattern. These segments can then be used to construct a new segment according to a given format. This processing is useful in the manipulation of natural language strings. Since the front end of the LISP 2 system will contain a finite-state machine as an input-string grouper and a syntax-directed compiler, the user will be able to change the language easily, specify new syntactic forms, define new data types, and make semantic additions to the language.

The compiler for LISP 2, first compiled with a bootstrap compiler written in LISP 1.5, will be written almost completely in LISP 2 itself; LISP 2 will thus be relatively independent of machine types, and we expect to create versions of it on a number of machines, such as the Q-32 at SDC, the IBM 360, the DEC PDP-6, and the GE 645. The compiler will work in a two-stage process. First, the external communication language will be translated into an intermediate language, which is compatible upward from the current LISP 1.5 notation. Then a list-structure assembly-language program will be generated and compiled by a simple machine-code assembly program. A preliminary version of the LISP 2 system should soon be in operation on the Q-32 at SDC.

Man-Computer Symbiosis - Warren Teitelman

A LISP programming system is being constructed to facilitate the development of sophisticated problem-solving systems. Emphasis is on easing the familiar debugging sequence: write some code, run the program, make some changes, write more code, run the program again, etc. As a system becomes more complex, making changes becomes harder and harder.

Our goal is to make the computer, via a programming system, play an active role in this modification process by providing the means whereby changes can be effected immediately, in ways that seem natural to the user. The flavor of the system is such that the user feels he is giving advice, or making suggestions to the computer about the operations of his programs; the necessary work to achieve the desired effect is performed by the symbiotic system. The system thus acts as an interface between the user and his programs.

A "format list-processing language", FLIP, has been developed and implemented. FLIP performs operations of the type done by COMIT, but is considerably more sophisticated and operates from within a LISP system. This language will be used to process the advice given by the user and will enable the symbiotic system to accept suggestions in a very relaxed format. It has already been used to construct a set of editing functions for LISP programs.

Consistent with emphasis on the importance of change and flexibility, the system itself may easily be modified by giving it advice about its own operation.

Mathematical Assistant - William A. Martin

I have embedded a picture-display language in the LISP language of the PDP-6. Now complex display programs, such as the ARGUS character-recognition scheme of W. Teitelman, are easily written. I have also developed a LISP program that displays mathematical expressions and equations. At present, the displayed equations have substantially the same form as those appearing in mathematical publications. This mathematical display program in PDP-6 LISP communicates with the rest of the mathematical assistant programs in the time-shared MAC LISP through an input/output dataphone system.

The mathematical assistant system is a new technique for the comparison and simplification of symbolic mathematical expressions. It uses finite-field arithmetic for hash-coding functions of a complex variable. By constructing a large but finite field (p=8, 589, 949, 373) within which there are representations of the integers and π, in an appropriate sense, one can use exponent arithmetic to map symbolic expressions onto integers. If two expressions are equivalent with respect to rational, exponential, and some trigonometric identities, they will, with high probability, map onto the same integer. In the recognition of common subexpressions, this system appears to be faster and more accurate than those which use canonical orderings.

Heuristics of Theorem Proving in Group Theory - Lewis M. Norton

We are developing a system of heuristics of theorem proving in group theory. It is coded in LISP and handles problems that involve the simple consequences of definitions. We are now using it to test methods of reducing irrelevant effort in proofs already produced. Rather than use explicit forms of the predicate calculus, it uses forms more similar to standard mathematical practice. Thus, the definition of abelian for a group G is written:

((ABELIAN G G) IMPLIES (AND (MEMBER A1 G) (MEMBER A2 G))
(EQUAL(*PROD A1 A2 G) (*PROD A2 A1 G))).

Since logical axioms are embedded in the flow of the program, the involved theorems can be proven more quickly than has been done by other methods. For example, the system proves:

Given: G a group, J a subgroup of G, H a normal subgroup of G.

Prove: J Ω H is normal in J.

In this work, we are trying to simulate the analytical and rational methods of human problem solving.

Computer Experiments in Finite Algebra - W. Douglas Maurer

Any set of mathematical methods and techniques may be mechanized on a digital computer. Even infinite mathematical structures may be studied, provided that the expressions used to speak about them are finite, since it is these symbols which the computer manipulates. The first mechanization is for finite algebra. This system for finite algebra consists of an English-like language with 80 commands for the manipulation of finite groups and semigroups, subsets, maps, and labeled constants. Rings and fields are to be added later.

The system accepts an arbitrary semigroup from the console, if the subgroup is given by its Cayley table (multiplication table). It can do the following:

1. construct the Cayley tables of four special kinds of group and three special kinds of semigroup;
2. take the direct product of any two Cayley tables and produce a third;
3. generate the subsemigroup of any semigroup generated by a single element or subset of that semigroup and then proceed to build the Cayley table of the subsemigroup;
4. find all the subsemigroups of a semigroup and all the normal and maximal subgroups of a group;
5. find all the left, right, two-sided, or only the maximal ideals of a semigroup;
6. test whether a Cayley table is associative and whether a semigroup has an identity, is a group, and is abelian;
7. tell whether a map from one semigroup is homomorphic to, one-to-one, or onto another semigroup;

8. print out, erase from its storage area, or rename any Cayley table, subset, map, and labeled constant;

9. add a zero element and a unit element to a semigroup;

10. produce the natural map of a group onto the set of left, or right, cosets modulo a subgroup, or the Cayley table of the factor group modulo a subgroup.

I have recently added a function to the system which determines whether any two given Cayley tables are isomorphic and, if so, demonstrates the isomorphism. This function runs in less than thirty seconds for two groups of twelfth order. I plan to introduce a program which accepts mathematical statements in a source language and produces object codes to verify or disprove the statements over a restricted class of test objects, such as a collection of counterexamples.

<u>A Chess-Playing Program</u> - Burton H. Bloom

An elaborate static-position evaluator has been substantially debugged for the CHESS program. It includes material balance, development, center-control, and space evaluations. Current plans are to also include king-safety and pawn-weakness and some other criteria. Several plausible-move generators are being tested. These look for captures for gain, checks, threads to capture for gain, retreating attacked units, and safe moves that avoid immediate loss of material. Plans are underway to provide for handling defensive moves, interposition, development, and center control.

Strategic heuristics are being debugged for complicating situations to exploit possible opponent's errors, and for controlling the use of the different generators in different situations. An elaborate system of Macro-operations is used for defining the evaluators and generators.

With debugging of current features, the program is expected to match low-to-average club players.

MATHLAB: On-Line Symbolic Computation - Carl Engleman (MITRE Corporation)

MATHLAB is a LISP program that is intended to provide the services of an on-line, time-shared computer to a mathematical scientist, such as a physicist, applied mathematician, or electronics engineer, for the daily performance of his tedious, mechanical, mathematical tasks.

In its present state of development, the program is primarily oriented towards symbolic, rather than numerical, computation. Employing simple commands, such as "solve" or "differentiate", the program has facilities for symbolic manipulations, such as addition, multiplication, and differention of expressions and equations. It is also capable of the simplification of expressions and the substitution of one expression within another.

Finally, it can, using the INTEGRATE program of M. Manove, succeed in the symbolic integration of certain expressions and the symbolic solution of certain expressions.

A detailed discussion of the MATHLAB program may be found in MITRE Corporation Technical Memorandum TM-04258, from Department 73, dated 26 May 1965.

INTEGRATE: On-Line Indefinite Integration - Michael Manove (MITRE Corporation)

INTEGRATE is a LISP program designed to symbolically integrate rational functions in one variable over the field of rational numbers. Any rational function is admissible as input, and such functions may be represented as the sums, differences, products, quotients, or integral powers of constants, polynomials, and other rational functions. The antiderivative of a rational function will be expressed as the sum of a rational function with rational coefficients, and the logarithm of a rational function possibly with complex coefficients.

The program will also integrate many rational functions of trigonometric functions, of the form f(sin x, cox x, tan x, cot x, sec x, csc x) where "f" is a rational function, but this feature is limited by the fact that answers are always expressed in terms of "tan (1/2)x".

The coding of INTEGRATE is divided into two major sections. The first section will find the rational part of the antiderivative and determine rational functions whose integrals are the summands of the logarithmic part. The second section will find the logarithmic parts of the antiderivatives of a broad range of rational functions. This, of course, is subject to the size limitations of computer memory and other hardware variables.

This program has been incorporated in the MATHLAB program of C. Engleman. A detailed discussion of INTEGRATE may be found in MITRE Corporation Technical Memorandum TM-04204, from Department 73, dated 22 April 1965.

BLANK PAGE

BIOLOGY DEPARTMENT

Molecular Model Building

Academic Staff

C. Levinthal

Research Assistants and other Students

P. Green	A. W. MacEwan	J. L. Sussman
R. Langridge	A. Pawlikowski	M. Zwick

Molecular Model Building - Cyrus Levinthal

Many of the recent developments in molecular biology have depended on an understanding of the three-dimensional structure of large molecules. Such an understanding is usually reflected in the ability to construct a three-dimensional model of a particular molecule. For really large molecules, however, such construction is difficult and time consuming and, if complete, would entail an impossible enumeration of the many small interactions which contribute to the molecular stability. Thus, if a structure is known, physical models can be built, but construction complexity prevents the use of such models in the examination of a large number of different configurations.

During the last six months we have written a set of programs with which we can construct, display, and analyse macromolecules on the MAC system's ESL display unit. With computer-controlled display and real-time control of the molecule, an observer can obtain a true three-dimensional visulization of a particular molecule; and by interacting with the program which generates the molecule, he can indicate the way in which particular parts of the structure are to be altered.

The first program written in this project calculates the coordinates of the atoms in a protein. Only those angles about which rotation is possible are entered as input variables in the program; all other rotation angles and chemical bond lengths are entered as rigid constraints. The calculation treats each chemical bond along the linear peptide backbone as a translation and a rotation of a coordinate system attached to individual atoms. In this way, the step from one backbone atom to the next involves a translation and a rotation-matrix multiplication. The updating of the rotation matrix is determined either by the fixed rotation angles, which are introduced as constraints in the program, or as input data which are added by an investigator. The basic data for these constraints on bond angles and lengths have been obtained over a number of years from the X-ray crystallographic studies of Pauling, Perutz, Kendrew, and many others, and they are now firmly based on experimental information.

The three-dimensional structure of the protein is determined entirely by the physico-chemical interactions which lead to the lowest energy configuration. Since the bulk of the free energy is contributed by many small interactions, which for the most part have extremely short range, we should like to determine for each pair of atoms whether the two members lie close together in real space and thus contribute substantially to the interaction energy. Since even a small protein molecule may contain 1500 atoms, the total number of pairs to be tested is very large.

To avoid the enumeration of all possible pairs, we have written a set of programs that first divides space into cubes and then makes a list of all atoms in each cube. By searching through the cubes, and listing those pairs in which one member is in a cube and a second member is in the same cube or in one of the twenty-six surrounding ones, we are able to enumerate all pairs of neighboring atoms. This list-processing procedure finds the holes in a molecule by locating empty cubes surrounded by filled ones; and, in addition, it determines the inside and outside of the molecule by finding cubes which have filled neighbors on one side and empty neighbors on the other side. The knowledge of these topological aspects of a folded protein is necessary for the assessment of interactions of particular amino acid residues with the water normally surrounding all protein molecules. These programs used SLIP subroutines written by Professor Weizenbaum, whose cooperation was invaluable in our work.

When an investigator wishes to alter the structure of a molecule, he inserts a pseudo-energy between a particular pair of atoms. An energy-minimizing routine, which looks at the sum of all interactions, including the pseudo-energy, calculates the appropriate change required for the variable angles. If the change leads to a violation of van der Waals contacts, another routine eliminates this violation. The latter routine, as well as those for energy minimization, require the calculation of the partial derivatives of the distances between pairs of atoms with respect to all of the variable angles which can effect these distances. We have used two procedures in altering the structure and eliminating van der Waals violation: the first uses one of several different minimization routines in a sequence of small steps; the second uses a linear approximation and calculates appropriate angular change by solving linear equations, subject to the additional constraint that the sum of all angle changes should be a minimum. When we have tested and run programs and subroutines together in a smooth system, we can attempt to solve for the unknown structure of a real protein.

By making use of molecular models constructed in this way, we can evaluate the importance of electrostatic interactions in the stability of protein configurations. These interactions occur because of the existence of a large number of small electric dipoles. Although each of the dipole interactions is small, they have a longer range of interaction than the van der Waal forces and are so numerous that they contribute appreciably to overall molecular stability. The results of our calculations indicate that dipole interactions produce a stabilizing energy, if the helical regions of the proteins are antiparallel to each other, and a destabilizing energy, if the helices are parallel to each other.

The structure of the myoglobin protein molecule has been completely determined by X-ray crystallographic methods. When we used the approximate coordinates of this molecule in our model, we found the net stabilizing effect of these dipole-dipole interactions to be significant. We have also used our model-building program to refine coordinates of the myoglobin model, by supplementing the X-ray data with data on the chemical bonds derived from other sources. Prior to this computer refinement, our attempts to carry out a calculation of this kind involved the construction and measurement of physical models made of brass wires. This calculation was both tedious and inaccurate for a molecule as large as myoglobin.

We have written a program that uses Project MAC's PDP-6 as a display unit for structures whose coordinates are generated on the 7094. This program was designed to test the usefulness, as aids in the visualization of a three-dimensional structure, of perspective and modulation of line intensity with depth. It is written in such a way that one can test the feasibility of a three-dimensional visualization scheme that requires fewer rotation calculations than the ESL display. These tests will determine the required characteristics of additional off-line display equipment.

We have written a set of programs which calculate the strain energy of displacement of chemical bond angles and lengths from their equilibrium positions. When the rigid constraints on bond length and angles in the programs described previously are relaxed, the number of interactions in a protein becomes prohibitively large. However, studies of small-molecule structure can be carried out in a reasonable amount of computer time. We have used this procedure to refine models of the sugars which are a part of nucleic acid molecules, and we are studying the effect of this on the overall structure of nucleic acids. Since we can carry out this refinement for any small organic molecule, we can, in principle, display the three-dimensional model of a small molecule when our only knowledge of it is what atoms compose it and which ones are covalently bonded together.

[Editor's Note: Figures 1 and 2 are stereographs of a DNA molecule and several protein molecules, that were photographed from the ESL Display Console at Project MAC. To use the stereo pairs without a viewer, ignore the double image print for the moment and focus your eyes on a point about 10 feet away. Bring the figure slowly up into your line of sight, maintaining the same focus of 10 feet. Four images will be within your field of view, and then the two middle images should merge into a central image. You now have three images side by side. The center one will be the stereo composite of the two printed images. You may need practice to hold this image until your eyes focus on the stereo image.

If you have difficulty with the multiple images, try holding your hand, edge on, in front of your nose, so that the left image cannot be seen by your right eye, and the right image cannot be seen by your left eye. This will form a crude stereoscope and allow you to concentrate on a single image, the stereo composite.

Figure 1 shows three views of the same portion of a DNA molecule. These were produced by photographing stationary images that differ by approximately 10 degrees of rotation about the Z axis. Normally, a person at the ESL Display Console is able to obtain a solid-appearing view of the molecule by real-time rotation of the image about three axes. The views shown in Figure 1 are approximately 10 million times life size.

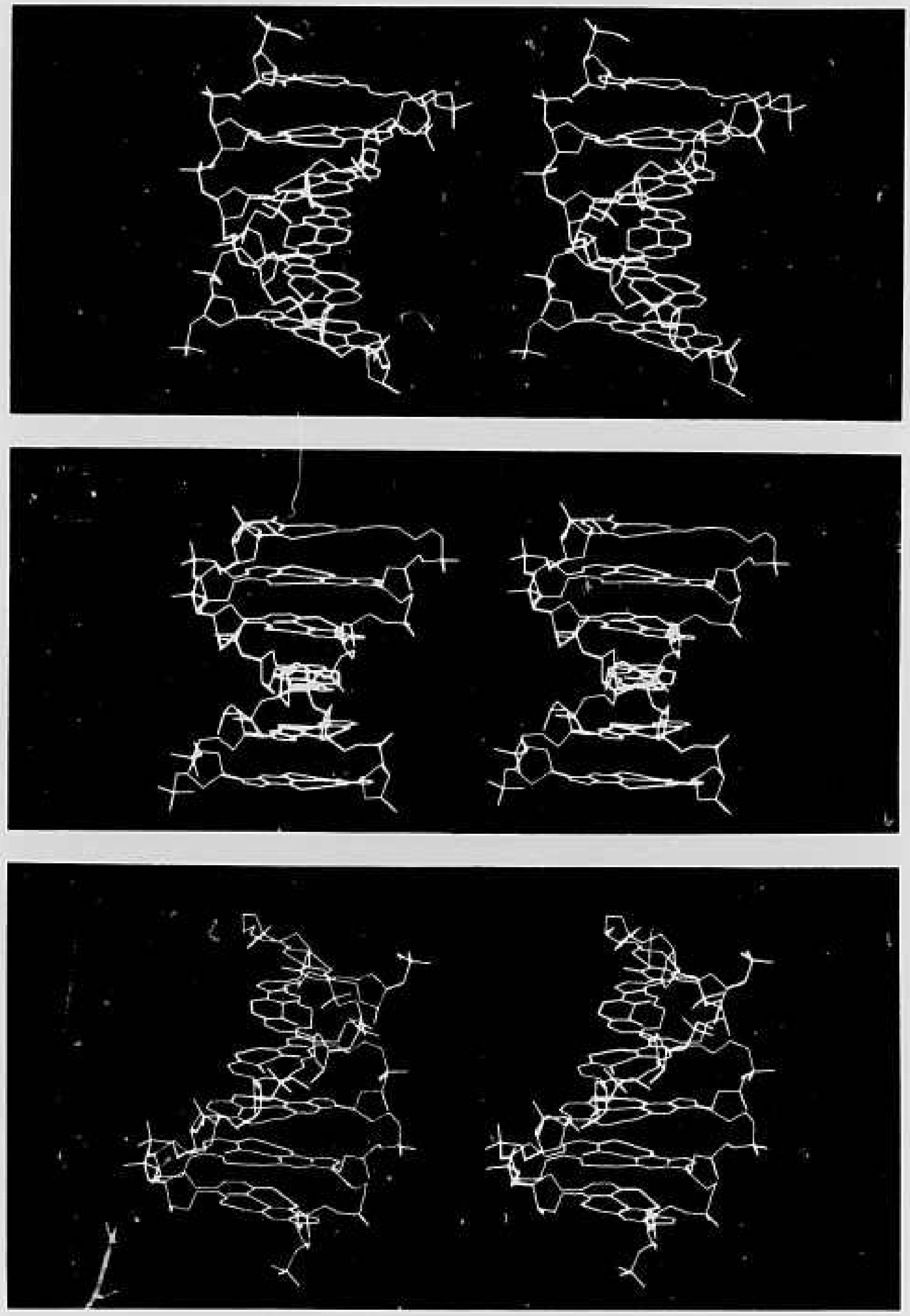


Figure 1. Stereo Pairs of a DNA Molecular Structure

Deoxyribonucleic Acid (DNA) is the carrier of genetic information in living systems, and consists of two phosphate chains that run in opposite directions and are coiled into a right-hand helix. Nitrogen bases, held together by hydrogen bonds, link the ester chains like steps of a spiral staircase and provide the sequence which carries the genetic coding.

Figure 2 shows three views of complete and abbreviated polypeptide protein chains arranged in the α - helix configuration. The α helix, capable of forming hydrogen bonds between all of its amide groups, is a substantial constituent of protein molecules, which perform vital enzymic functions, such as controlling chemical reactions, in living systems.

The top view in Figure 2 is a fairly complete polypeptide structure, while the other two views are simplified versions of the same molecule. The bottom image is the same molecule as the middle one, showing the exaggerated stereo separation obtained with approximately 20 degrees of rotation.]

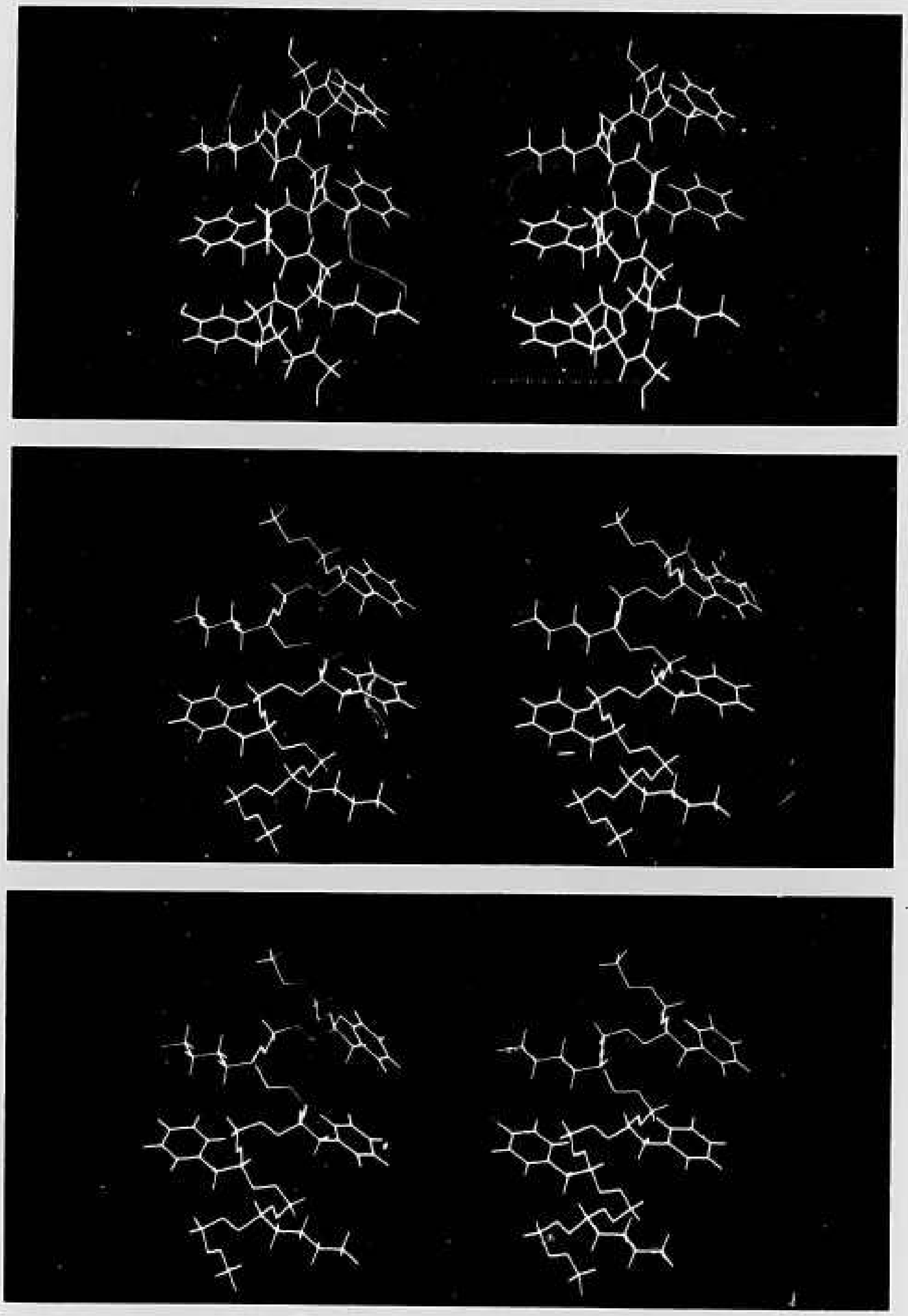


Figure 2. Stereo Pairs of a Protein Molecular Structure

BLANK PAGE

CIVIL ENGINEERING DEPARTMENT

Computer-Aided Teaching

The Structural Design Language

Dynamic Structural Analysis

Soil Engineering Problem-Oriented Language

I/O System Research

Bridge Design

Optimal Synthesis of Road Networks

Academic Staff

J. M. Biggs
J. F. Brotchie
C. A. Cornell
A. G. H. Dietz
A. M. Hershdorfer
T. W. Lambe
R. D. Logcher
C. L. Miller
J. M. Roesset
D. Roos
R. L. Schiffman
G. M. Sturman
J. Sussman
J. R. Walton
J. W. Weber
R. V. Whitman

Non-Academic Research Staff

A. T. Foster
R. V. Goodman
J. M. Hodnick
K. Reinschmidt
R. A. Walter

Research Assistants and other Students

T. H. Asselin
W. Bailey
L. N. Beckreck
D. Beltran-Maldonado
J. R. Brack
J. E. Daily
R. L. Daniels
R. E. Efimba
H. C. Frazier, Jr.
D. Gardner
G. Guzman Baron-Torres
R. A. Hadley
W. H. Linder
S. Lipner
S. G. Mazzotta
W. M. Pecknold
J. R. Roy
L. C. Teague
B. J. Vander Molen

Computer-Aided Teaching - Daniel Roos

The Department of Civil Engineering has used the Project MAC time-sharing system in its undergraduate and graduate education programs. Three different types of educational activities relied heavily upon CTSS: classroom demonstrations, homework and term paper assignments, and an experimental time-sharing section of course 1. 15.

The time-sharing system has been used very successfully in classroom demonstrations to assist in the teaching of engineering. In a classroom equipped with closed-circuit television, an IBM 1620 computer, CalComp and Gerber plotters, and an IBM 1050 remote console connected to the Project MAC computer, the instructor has all the computing facilities available to demonstrate points that he is making. If a student asks the perennial question, "What happens if______?" the teacher can immediately demonstrate the implications using on-line computer facilities.

Several problem-oriented languages have been developed, with which an engineer can communicate with the computer in engineering terminology. Two of these languages, STRESS for structural problems and COGO for geometric problems are being used in conjunction with time-sharing for educational applications by both instructors and students. A student does not need intimate computer knowledge to effectively use these languages. With minimum instruction, generally one hour or less, he can use the problem-oriented language and the time-sharing system to obtain solutions to engineering problems.

The Structural Design Language - Robert D. Logcher, Gerald M. Sturman, Richard V. Goodman, Lewis C. Teague, Salvatore G. Mazzotta, and John R. Roy

Structural design, the determination of structural elements and their layout and makeup, is a task that involves innumerable variables and decision parameters and requires extensive computation and decision-making. The nature of the computations, and the manner in which they are executed, makes classical forms of computer use inadequate and unsatisfactory. The purpose of the work described here is to determine the form and organization required for a useful design system.

Structural design requires interactions with other technologies involved in the design product, such as mechanical and electrical engineering and architecture. The engineer must accept and recognize the conflicting aims of these technologies and evaluate compromises analytically. He is therefore engaged in an iterative process in which he must remain flexible. The iteration consists usually of a series of trial solutions together with an intuitive searching process that leads to an optimum solution.

Generally the amount of computation performed between decisions is small, but varies greatly with the problem, the engineer, and the various parts of the design task. Problems vary widely in size and complexity, from a few hundred variables and parameters to thousands. Depending on the problem and the engineer's sophistication, the design task may be automated with all decisions made prior to solution or may be broken into many processes with decisions inserted during solution. A process of design such as elastic analysis involves a great deal of computation; while others, such as checking performance against criteria, require very little.

A time-sharing environment is required to join the creative design process to a computer. The user may thus perform small and large tasks in an unpredetermined order and express decisions between tasks. The problem-oriented language STRESS represents the first phase in computer-aided structural design. This language contains a data-input mechanism and programs for linear elastic-frame analysis. Since this analysis is the most difficult computational problem, but the best defined mathematically, its solution was the first undertaken. General data-structure capabilities for engineering data were developed for STRESS which make its logical inclusion in a broader language possible.

Over the past year, the Structural Design Language (STRUDL) has been developed in an open-end form. The major commands of this system are:

1. PROBLEM INPUT
2. PRELIMINARY ANALYSIS
3. DETERMINATE ANALYSIS
4. MEMBER SELECTION
5. STIFFNESS ANALYSIS
6. SELECTIVE OUTPUT
7. WEIGHT DETERMINATION
8. CHECK PROCESS
9. SPECIFICATION CHECK
10. REVISE TABLES

Each of these calls a particular operation commonly used in the design process. The engineer can decide which operations are to be used and in what order they are to be executed.

For example, PRELIMINARY ANALYSIS permits him to insert assumptions regarding the behavior of an indeterminate structure and thereby to make the problem determinate. He may then use DETERMINATE ANALYSIS to obtain a complete solution. MEMBER SELECTION, based on previous-design-criteria input and permanently-stored tables of standard members, may follow, and in turn be followed by STIFFNESS ANALYSIS, which provides a rigorous solution for the indeterminate structure. With SELECTIVE OUTPUT, the user can obtain only that output which is of immediate interest, rather than voluminous data which is actually generated.

As modifications to the structure are made, either automatically or at the user's judgment, new analyses are executed until the engineer feels that an optimum design has been achieved. This type of on-line operation provides a new dimension to structural design and gives the engineer an ability for optimization not previously possible.

Seven of the commands listed have been coded and implemented. The remaining three, as well as others which further extend system capability, require further work.

Dynamic Structural Analysis - J. Melvin Biggs and Daniel Beltran-Maldonado

In the design of Civil Engineering structures, the problem of analyzing those subjected to dynamic actions arises frequently. The laborious nature of the analysis on the one hand, and the possibility of more refined methods on the other, make the solution of these problems by a computer very desirable. Indeed, the technical literature indicates extensive use of computers in this connection, mostly through special-purpose programs coded to solve individual problems. The development of a programming system to analyze a variety of structures for free or forced vibration, such as space frames, space trusses, plane grids, plane frames, and plan trusses, is in progress. This system is a problem-oriented language called DYNALS, similar in many respects to STRESS. DYNALS is unique, not only because it is applicable to a wide range of structural configurations, but also because it encompasses a large part of the total design problem. Given the geometry of the structure and the stiffness and mass of individual elements, it forms the total mass and stiffness matrices, obtains eigenvalues and eigenvectors, and computes dynamic response of the structure in minute detail for any specified distribution and time variation of applied forces.

Three aspects of the problem make implementation in the time-sharing mode highly desirable:

1. The iterative nature of the design process calls for a simple and rapid way of modifying a problem and analyzing the effects of such modification on the structural response,

2. The wide variety and great volume of possible output available from a dynamic analysis make the ability to request at will selected forms of output very desirable,

3. Time-sharing offers unique advantages in the debugging and testing process, which are quite valuable to a user who is working with such a large and complex program block.

DYNALS will be incorporated into STRUDL, which deals with a broader portion of the design process for Civil Engineering structures. To make the system efficient for both large and small structures, the various data arrays are dynamically allocated so that only needed arrays are required in core at any given time.

DYNALS accepts and decodes data pertaining to the continuous structure and performs a "discretization" of the stiffness and mass characteristics. The discrete system consists of a number of degrees of freedom, equal to the number of kinematic degrees for a joint, times the number of joints. The analyst, presented with various alternatives regarding the formation of the mass matrix, chooses with regard to the nature of the problem at hand. He may accordingly reduce the number of degrees of freedom to a specified selected set. A wide variety of output, available upon request from the console, ranges from eigenvalues and eigenvectors of the free vibrating structure to the complete response of one and all of the mechanical or geometrical quantities of the structure while it vibrates under a time-varying load system.

DYNALS may be subdivided into the following programming blocks:

1. INPUT phase, decoding of input statements, storage of data and process parameters, and consistency checks,

2. Formation of the logical stiffness matrix,

3. Formation of the logical mass matrix for the case where member masses are specified,

4. Reduction of the number of degrees of freedom to a selected set when specified,

5. Reduction of loading data,

6. Solution of the eigenvalue problem,

7. Using the characteristic shapes obtained in 6. as input joint displacements, the preparation of sets of static solutions for forces, reactions and member distortions, (n is the number of modes to be combined.)

8. Using the results from 5. and 6., the numerical integration of the modal equations of motion, the results of which are used to combine the static solutions developed in 7., (Record is kept of maxima and time of their occurence for forces, reactions displacements and distortions.)

9. OUTPUT phase.

As of June 1965, the system design and a substantial portion of the programming is complete.

Soil Engineering Problem-Oriented Language - Robert L. Schiffman and Laurence N. Beckreck

SEPOL (Soil Engineering Problem-Oriented Language) is a system of interrelated programs which serves a three-fold purpose. First, SEPOL is designed as a communication system in which the communication between man and machine is in an interrogation form that elicits the program from the user and channels the program towards the desired objective. Second, it is a computer-aided design system that so widens the area of choice and breadth of calculation that by sheer speed and utility, a greater range of calculation is open to the soil engineer. Third, it performs a basic study of the nature of soil engineering analysis. By subdividing the system and delineating the areas of common logic, it identifies the basic hypotheses and analytical units.

SEPOL I, covered in this report, is a system designed to calculate, via time-sharing, the magnitude and progress of settlement of an earth mass, when the soil surface is subjected to a specified loading. It was designed as a prototype to determine the feasibility and usefulness of this type of computing system, and also to provide useful information for soil engineers.

SEPOL I is composed of many small subroutines, each of which represents a unique task that is always performed as a unit and can be performed anywhere and many times in an analytical sequence. The control function of SEPOL I is interrogatory. A set of questions is posed; and, since the response to the questions provides the linkage between boxes, the set of questions and answers builds, for each situation, a specific main program. SEPOL I was designed as an easily-expandable system. To add a new method of analysis requires first the programming, in FORTRAN, of tasks or calculations not already in the system, and then the augmentation of a dictionary of variables. Access to the new system is accomplished by adding choices at the appropriate decision points.

I/O System Research - Ken Reinschmidt, D. A. Gardner, and John W. Weber

The hardware which connects the IBM 1620 to the MAC time-sharing system consists of two logically-different parts. First, a general I/O interface to the 1620 makes it easier to connect non-IBM devices, and second, a dataphone interface which goes between the general interface and the dataphone which is connected to MAC. Because of certain idiosyncrasies in the 1620 I/O control, such as the lack of interrupt, time-sharing operation must be under software control. Subroutines, written in 1620 assembler language and available to the FORTRAN programmer, allow him to use the 1620 typewriter and/or card read/punch as a CTSS console, pass numeric data between programs in 7094 and the 1620, pass alphanumeric data, and perform alphanumeric comparisons on data in the 1620. The subroutines were developed in a modular form which facilitates modification or expansion of the present system. Mr. Weber also developed a supervisor subroutine for the 1620, which gives a more automatic operation. The main uses of the system have been for obtaining plotted results of data developed at the ESL Console and for card I/O of lengthy I/O operations.

Another improvement on the 1620 was made by the addition of the Gerber VP-600 plotter as an output device. This plotter was put on-line with the 1620 in August, 1964, through a modification of the old CalComp Plotter. Several plotting subroutines were developed for the Gerber unit: line, number and character plotting routines by Mr. Gardner; and then curve and circle routines.

An on-line, date-acquisition system has been developed jointly by members of the System's Laboratory and the Structural Model's Laboratory. The hardware part of the system, connected to the 1620 through the general interface mentioned above, consists of electro-mechanical switches, digital logic, signal conditioning and amplification, and an analog-to-digital converter. The system allows the 1620 to switch to any one of 50 strain gauges and read the converted voltage into storage.

A large set of software, which facilitates the use of the system, is a problem-oriented language for the data acquisition system. With this language, the user can do any of the following:

1. Balance any or all gauges,
2. Select how many and which gauge he desires to read,
3. Read each gauge as many times as desired,
4. Do simple calculations,
5. Convert his data to several different forms,
6. Plot results,
7. Store intermediate results on the disk,
8. Change loading and repeat operations.

The system was developed by Mr. Reinschmidt, has been used in classroom demonstrations and on some plate studies, and is now being used extensively on models of a dam under construction in Spain.

Bridge Design - John F. Brotchie, Charles A. Cornell, Jose M. Roesset, and Guillermo Guzman Barron-Torres

The purpose of this research is to develop an integrated and comprehensive system for the directed design and drafting of highway bridge structures. It is similar in several respects to the Structural Design Language reported elsewhere, but is oriented specifically toward the design of highway bridges.

The design system proposed consists of a series of logical blocks, self-sufficient in themselves. These blocks, each one consisting of a large and expandable number of links, are to be interconnected and each is amenable to direct input and output. Individual blocks perform the following:

1. An internal definition of geometry;
2. A determination of suitable alternate forms, span arrangements and materials;
3. A design of superstructure, piers, abutments, approaches, or parts thereof;
4. An analysis of the structure or components above;
5. Optimization;
6. Detailing;
7. Cost analysis and calculation of quantities to various degrees of detail, depending on the stage of the design at which the link was entered;
8. Documentation, including drafting and specification preparation.

The point of entry to the system, the flow between the blocks, and the kind of exit may be chosen by the designer at the beginning of the process, but are preferably chosen during it. Time-sharing facilitates this choice, and a large computer facility provides the storage necessary to retain all required information throughout the process and communicate it to the designer in the most efficient way, whether by plotter, printer, teletype or oscilloscope. Thus, the logic of the design process and the basis for determining design parameters is controlled by the designer.

Some phases of the system are already completed and were recently linked by Guillermo Guzman to demonstrate a time-sharing version of the program. These were the input phase, the definition of internal geometry, the selection of alternates for straight bridges, and the design of simple-span, composite, precast-prestressed, concrete superstructures together with their optimization on the basis of a comprehensive minimum cost. A pilot version of the steel design phase will also be added shortly.

Optimal Synthesis of Road Networks - Alan M. Hershdorfer

This study was originally an investigation into the properties of optimal traffic flow patterns on metropolitan road networks. A total-travel-time function to be minimized is expressed as a piecewise, linear, convex approximation to the experimental travel-time vs. traffic-volume relationship. The linear programming code RSMFOR was used to determine cost flow solutions for a multi-commodity flow system. The model was then extended to a mixed-integer programming formulation. The integer variables represented decision variables related to the provision or deletion of road capacity. In this model, numerous related problems of traffic network synthesis are studied. The function to be minimized is a linear function of construction cost and user cost.

The simplest synthesis problem, that in which construction cost is nil, occurs in the problem of optimally orienting a set of one-way streets. Integer-linear constraints on lane allocation require all lanes in a road link to be oriented in one of the two directions or to be split equally between the two directions. This one-way street problem was studied extensively with a time-sharing version of the Land and Doig Algorithm for mixed-integer programming, which prescribes values for integer variables in an ordered sequence of linear programs. With time-sharing, the algorithm was monitored during computation and, thus, trial solutions were very rapidly evaluated. Since the structure of the flow constraints forced certain variables to near integer values after a small number of integer decisions had been made, a good primal feasible integer solution was very rapidly determined. Time-sharing, used as a monitor on the algorithm, has enabled us to develop a heuristic method for quickly obtaining solutions to problems that involve the synthesis of flow networks.

COMPUTER COMMUNICATION STRUCTURES

Sequences and the Four-Color Problem

A Design Language for Digital Systems

Man-Machine Communication

Research on the Theory of Automata

Program Segmentation

Waveform Transformation and Graphical Display

A Table-Directed Translator

On-Line Braille

Analysis of Time-Shared Computer Systems

Queueing Models for File Memory Operation

Input/Output Subsystems

Non-Repeatability of Multi-Process Computations

Semantics for Multiprogrammed Computations

Optimal Allocation of System Resources

Automatic Flowcharting

Visual Information Processing

Academic Staff

D. A. Huffman
J. B. Dennis
E. L. Glaser
R. Y. Kain
D. S. Kuck
C. L. Liu
R. McNaughton
C. E. Shannon
T. G. Stockham, Jr.
H. M. Teager
J. Weizenbaum

Non-Academic Research Staff

S. Lorch - Massachusetts General Hospital
M. T. McGuire - Massachusetts General Hospital
G. C. Quarton - Massachusetts General Hospital
A. Rutchka

Research Assistants and other Students

A. Buckles
G. J. Burnett
R. H. Campbell
G. D. Y. Chang
P. Denning
H. Ellis
R. C. Gammill
D. A. Lasher
H. F. Ledgard
F. L. Luconi
H. G. Murray, Jr.
A. S. Priver
L. J. Rotenberg
A. L. Scherr
H. S. Schwenk, Jr.
D. H. Slosberg
A. A. Smith
E. C. Van Horn, Jr.
S. A. Ward
E. Westerfeld
D. U. Wilde
S. Zucker

Sequences and the Four-Color Problem - David A. Huffman

Methods for the generation of binary-valued pseudo-random sequences, which are more general than shift-register sequences, have been studied. These sequences have the interesting property that they remain invariant to a type of transformation which, in a manner reminiscent of tag systems, substitute one string of symbols for another.

The four-color problem (a famous, as yet unsolved problem in topology) has been studied with the view that it furnishes a difficult and realistic example of a problem for which a man, aided by appropriate program-generated computer displays, may be able to gain insight more rapidly than a man working with pencil and paper. The maps associated with this problem are nicely suited to representation by certain simple, symmetric list structures.

A Design Language for Digital Systems - Gerald J. Burnett and Jack B. Dennis

This research was promoted by the need for a more consistent and convenient method of designing digital systems and by the desire for computer aid in this design process. We have developed a design language to meet a number of requirements. Some of the more basic requirements are the following:

1. The language must be complete, that is, it must be able to describe the logic and timing of any digital system;

2. The language must be convenient for the designer and must be simple enough to express all of the common hardware logic and timing functions;

3. The language must have features analogous to macros and repeats in assembly languages in order to describe readily-iterated logical structures;

4. The language must be able to describe a system in sections or components that are designed by different individuals.

As an example of the description of a familiar logical structure in the design language, we describe the logic required to perform parallel binary addition of a number contained in register B to a number contained in register A and leave the result in register A.

The addition method used for the example involves two steps. When the pulse named "partial-add" occurs, each bit of register A is complemented if the corresponding bit of register B is one. The pulse named "carry" complements a stage of A if there is a carry signal into that stage. The description of one stage of the adder is as follows:

```
begin partial-add:   B[ i ]  ↑ A[ i ] ;
      carry: if C[ i ]  then  ↑ A[ i ] ;
      where: i ≠ 0 then
          C[ i-1 ]  = C[ i ] ∧ A[ i ] ∨ ¬ C[ i ] ∧ B[ i ] ;
end;
```

The combinational logical operators ∧, ∨ and ¬ have their usual significance, while the upward arrow denotes the action of complementing, as a unary operator, or of conditional complementing, as a binary operator. The first two statements of the example show ways of indicating that certain actions are to take place on the occurrence of definite pulses, either absolutely as in the first statement or conditioned by a logic level as in the second. The third statement illustrates several features of the language. The where clause indicated that the statement following then is to be considered part of the description only if the Boolean expression $i \neq 0$ is true. In this instance, the following statement represents combinational logic that generates level C[i-1] according to the expression at the right of the equivalence sign.

The complete description of the adder involves iteration of the above logic for each position in the two registers and is handled by a for statement as shown below. Certain declarations are included so that the types of named objects are unambiguous. The integer quantity n is the number of bits in registers A and B and is presumed to be assigned a definite value outside this portion of the description.

```
begin
        integer n;
        register A[ 0, n-1 ] , B[ 0, n-1] ;
        level    C[ 0, n-1 ]
        add:     partial add;
                 delay 500n;
                 carry;
        C[ n-1 ] ≡ 0;
        for i = 0 through n-1 then
        begin partial add:  B[ i ] ↑ A[ I ] ;
              carry: if C[ i ] then ↑ A[ i ] ;
              where  i ≠ 0 then
                     C[ i-1 ] ≡ C[ i ] ∧ A[ i ] ∨ ¬ C[ i ] ∧ B[ i ] ;
        end;
end;
```

The pulse add performs the partial addition and initiates a 500-nanosecond delay. At the end of the delay, the carry pulse occurs. The meaning of the for statements is intrepreted as if the (compound) statement following then were written out successively for integral values of i from 0 through n-1.

Work related to the design language will be devoted next to the development of suitable simulation techniques and means whereby a designer can employ the language on-line in the process of creating digital system plans.

Man-Machine Communication - Joseph Weizenbaum

SLIP

SLIP, a list-processing system capable of being imbedded in a large class of algebraic compiling languages, was imbedded in the MAD system. MAD is the compiling system most frequently used in CTSS, and therefore, SLIP has been made available to large sections of the MAC user's community. A number of students and faculty members have made use of SLIP for research and thesis purposes.

OPL

The on-line SLIP-based programming system OPL was rewritten in the MAD-SLIP system discussed above. It has served as a test bed for a number of experiments, particularly text manipulation, and has been an invaluable tool in clarifying issues underlying interactive programming systems, their design and implementation. The problem of the programming system-user interface has been illuminated, but not solved.

ELIZA

A major effort in the design of a natural-language, on-line, man/machine conversational system was begun and carried forward during the reporting period. At this time, ELIZA is capable of conversing with users in natural language in certain restricted contexts. We are beginning to expand ELIZA so that it can form the basis for both information gathering and a retrieval system operating on a natural-language basis. ELIZA was also used for experiments on two-person conversational interaction at the Massachusetts General Hospital. The development of ELIZA required implementation of powerful text manipulation functions within MAD-SLIP. These latter are of general use as external MAD functions.

Research on the Theory of Automata - Robert McNaughton

A. CLASSIFICATION AND COMPLEXITY OF FINITE-STATE OPERATIONS

The most commonplace distinction in the theory of automata has been that between finite-state operations and operations requiring potentially-infinite automata. Partial motivation of our research has been the contention that finite-state operations should themselves be classified in some interesting manner. Stimulated by some work of M. P. Schutzenberger, we have proposed a notion that we think gives an important classification, namely the notion of a "topological" event (a term which we do not feel is optimal).

Interestingly, the notion can be characterized in several distinct ways: 1) as an event whose reduced-state graph contains no loop that counts any word modulo m, for m $\geq$; 2) as an event expressible as a regular expression containing free use of all the Boolean operators, but without star (denoting closure, or iteration); 3) as an event expressible in a certain first-order language of symbolic logic; 4) as an event realized by a nerve net in which feedback is confined to the case in which output of a neuron drives one of its own inputs; and 5) as an event whose semi-group contains no non-trivial subgroups. The fact that this class of events can be characterized in so many different ways - structurally, behaviorally, and algebraically - each interesting in its own right - is evidence enough of the importance of the concept. A long paper on this matter is planned, which will include proofs establishing the equivalence of the five characterizations.

There are two interesting ways of defining the loop complexity of a regular event, each in terms of a class of regular expressions. The first class is that of restricted regular expressions, i.e. those made up from the operators union, concatenation, and star. The second class consists of those made up of these operators and, in addition, intersection and complementation; such expressions are called "general regular expressions". Thus general regular expressions contain all the Boolean operators, while restricted regular expressions contain only union. In each case, the loop complexity of an event is defined as the minimal star height of all regular expressions in the class that represent the event. (The star height of a regular expression is the maximum length of a nested chain of the stars.) Thus, for example, the loop complexity of a topological event with respect to general regular expressions is zero. But there are topological events with arbitrarily large star height with respect to restricted regular expressions.

It seems that the most useful tool for investigating star height with respect to restricted regular expression (which is the problem studies originally by L. Eggan) is graph theory. On the other hand, the most useful tool for investigating star height with respect to general regular expressions is the theory of finite semigroups. Both concepts seem interesting and important, but both have unsolved problems. Thus, we do not even know whether there exists an event of star height 3 with respect to general regular expressions; and, although we do know the existence of events with arbitrarily large star height with respect to restricted regular expressions, we do not know whether there exists an algorithm for determining this star height when a regular event is given.

B. CONSTRUCTION OF WELL-TIMED NETS FROM BADLY-TIMED ELEMENTS

Research in this area has been a continuation of research at the University of Pennsylvania. A seminar was held consisting of some members of the Machine Structures Group of Project MAC and some design engineers from Honeywell. The number attending was small, but the discussions were intensive and fruitful. Some improvements were made in the report written at the University of Pennsylvania (Badly-Timed Elements and Well-Timed Nets, by R. McNaughton). This report will be issued as a monograph next year.

C. INFINITE HISTORY OF FINITE-STATE MACHINES

Two widely-differing studies in the literature have led into this area. J. R. Buchi studied a certain system of symbolic logic, and developed a decision procedure by showing, in effect, that every formula in the system says something about the infinite history of a finite-state machine. And D. Muller analyzed the behavior of an asynchronous circuit by regarding the circuit, which is subject to varying input conditions, and in which the relative timing of the parts is unpredictable, as a finite automaton with an infinite duration in time: a buzzer, for example, would be conceived of as a two-state automaton going from one state to the other ad infinitum.

The proof of an analogue to the so-called Kleene Myhill theorem for the infinite-duration problem has been written up, and will be submitted for publication. Also, there is an interesting connection between this problem area, and the topic of infinite games; i.e., games which two players play in turn an infinite number of times.

It would appear that this problem area is more abstruse than the other problem areas. In spite of the fact that the area was approached by two widely separated studies, no new application seems to be forthcoming. We are far from the position that this area is worthless; however, we do think it deserves less attention than the other two.

Program Segmentation - Richard Y. Kain and David J. Kuck

A question of major importance is segmentation of users programs to decrease memory space and swap times. There are several problems which immediately arise. The major one relates to efficiency of the algorithm which might be developed to perform such segmentation. In particular, is the time spent segmenting a program greater than the time saved?

Four approaches to this problem are:

1. The compiler examines a program's gross structure and makes simple decisions about where to divide the program;

2. The compiler divides a program into small "atoms" and then attempts to combine the "atoms" into segments;

3. The compiler divides programs into small segments, and then adds to each segment a small program which determines those segments which might be brought into memory at the same time;

4. The compiler describes possible paths through a program in terms of regular expressions, which are examined when allocation decisions are made.

The first approach is difficult to implement without some subdivision of a program; therefore, it will be considered as a special case of the second method.

The second method "reduces" the segmentation problem to a clumping problem: given a set of atoms, which are most closely related to each other? A number of approaches to this problem have been considered, but all unfortunately assume that there is some way to measure "interrelationships" between pairs of atoms; thus reducing the interrelationship to a number. We will assume that these numbers have been placed in a matrix, A, where A_{ij} is the interrelationship between atoms i and j. Some clumping strategies are:

a) Build clumps by adding "most suitable" neighbors until some criterion of maximum size or minimum suitability is met;

b) Partition matrix A until the sum of the elements not in the matrices along the main diagonal is small;

c) Take random walks through the "atoms," letting the probability of a path be proportional to the interrelationships of the elements at the ends of the path, with termination by length of the walk;

d) Find the largest eigenvector, whose components are the probabilities of visits to the corresponding states in the long run, and clump according to the size of the components of the eigenvector.

The third approach is a crude attempt towards "dynamic segmentation." By including a simple program which determines neighbors (possibly by examining the parameters of the running program, such as loop limits) with each atom, only those atoms which are likely to be used during the next quantum need be fetched. If the decision is too complex, a standard set of atoms would be used.

The fourth approach is another attempt, less sophisticated than the third, to dynamically segment the program. Program data are not checked in the choice of atom set and less economy is achieved. This method is superior to a priori segmentation, since the beginning atoms in a segment might not be needed if the previous execution had halted in the middle of a segment.

Progress has been made in finding interrelationships among methods 2a through 2d, though no useful generalizations are immediately possible. Unfortunately, evaluation must depend upon detailed experience with each scheme or simultations which are necessarily very dependent upon the models of user programs.

Waveform Transformation and Graphical Display - Thomas G. Stockham, Jr.

A subroutine for the plotting of numerical data on the E. S. L. display was completed by Henry Ledgard. The plotting process gives freedom from details of scale-factor selection and, at the same time, produces graphs which are labeled decimally in convenient round numbers and may have logrithmic or linear scales on either axis. Subprograms that perform various subtasks for the logical specification of completed plots are available as a separable package. These routines are free from constraints imposed by the specific display unit for which they are intended and, thus, provide a general service for any plotting equipment. Their main subtasks are generation of numerically-convenient plot boundaries and associated scaling parameters, generation of normalized grid-line parameters, generation of text for dimensioning, and scaling of data to fit the normalized plotting space. Finished plots are produced from normalized, floating-point array pairs of arbitrary length. The total generation time is between one and two seconds on the IBM 7094.

An algorithm has been developed by W. R. Chiodi with sufficient flexibility to implement a variety of linear transformations, such as Fourier, Hilbert, and convolution transformations. H. G. Murray, Jr. has realized the proposed algorithm in a computer program. This system operates on a standard-parts technique by which a required variability is concentrated in two building-block functions, two simple sets of rules for combining them, a set of parameters, and a simple transformation on these parameters. Since these variables can be specified through suitable numerical tables and brief sets of algebraic statements, the bulk of the computation is provided by a static framework into which they are embedded. This work will become part of a problem-solving system in electrical linear-system theory. (See Chiodi, and also Murray, Appendix B.)

A Table-Directed Translator - Chung L. Liu

Research in the table-directed translator system has continued for the past year. In order to further investigate and evaluate some of the ideas conceived in the course of our research, such a system is being designed and implemented.

It is also hoped that this system, when completed, will extend a great deal of our programming powers. While a table-directed translator system provides users with the flexibility of designing their own programming languages, it is most important from the users' point of view to learn to use it with ease. This is what we stress in our design. Clear boundary lines between modules of the system and simple input languages are emphasized.

It is expected that most parts of the system will be programmed in a higher-level language, so that reprogramming work, due to hardware transition, will be minimized. We intend to use the MAD language for this purpose.

On-Line Braille - Edward L. Glaser

It should be understood that Braille is not a simple one-to-one code with normal characters of the English language. Rather, it is reminiscent of shorthand in that many standard contractions and abbreviations are used. Although these contractions and abbreviations are not phonetic, the rules for their use are often context dependent. As a consequence, a computer is necessary to convert standard teletype code to a modified teletype code that is adequate to drive an electric Brailler.

Although this could have been done within the IBM 7094 computer at Project MAC, it was decided to use a separate satellite computer, so that Braille would also be available in the earliest phase of the new MULTICS time-sharing system on the General Electric 635 computer. As a consequence, a Digital Equipment Corporation PDP-8 computer was acquired this year, since it is a small, high-speed, versatile machine that is completely adequate for the job. To produce Braille output, we selected an electric Braille Writer that was designed and developed in the M. I. T. Mechanical Engineering Department.

The program for the conversion from teletype code to Braille has been written and checked out. Also, the interface between the PDP-8 and a Data Set has been checked out so that the Braille system can communicate with the MAC system as though it were a standard teletype station. Still remaining are final check-out of the interface between the PDP-8 computer and the electric Brailler and check-out of the Brailler itself. (See Luconi, Appendix B.)

Future plans include use of the PDP-8 machine as a base for additional studies of man-machine communication. Some of the possible areas of investigation are: auditory coding, interaction of auditory and visual stimuli, and visual and tactile stimuli. To date, a method for producing arbitrary auditory wave forms has been designed and fitted to the machine. It consists of two 6-bit digital-to-analog converters that are driven by the machine. The output analog wave form is sent through low-pass filters to stereo amplifiers and loud speakers. Programming the equipment for this application has not been completed.

Analysis of Time-Shared Computer Systems - Allan L. Scherr

Some aspects of the operation of time-shared, interactive computer systems were analyzed by making extensive user and performance measurements of the Project MAC system. Emphasis was on the reaction of hardware systems to the demands that users make. Simply stated, the problem was to characterize both time-shared systems and their users in order to predict the performance of the two operating together.

Portions of the problem included specification and measurement of user characteristics, the development and verification of both simulation and mathematical models for time-shared systems, and the specification and measurement of performance metrics for such systems.

First, simulation models were used to study the effects of changing small details in the operation of CTSS-like systems. Then, a continuous-time Markov process model was derived to predict the performance of a broad class of systems. Three lengthy programs were written to gather CTSS data. The first measured interaction parameters; the second measured input, output, and idle times of the system; and the third, statistics for interaction parameters of five system commands. This data was gathered on the MAC system between October 1964 and March 1965.

The CTSS data were used as a basis for comparison with model predictions, and, in order to take measurements and build models, many definitions of commonly used time-shared terminology were made precise. This research was used as the basis for a doctoral thesis. (See Appendix B and MAC-TR-18.)

<u>Queueing Models for File Memory Operation</u> - Peter J. Denning

A problem which has received much attention at Project MAC is the scheduling of user programs. The Scheduling Algorithm assigns each waiting program a burst of processor time during which it runs; and each program continues to receive bursts, at intervals, until it has run to completion. While not running, it is stored on a drum or disk. The process of placing a program in core memory or of removing it from core and placing it on the drum or disk is called swapping. An optimum Scheduling Algorithm results in the most efficient system operation and yet treats the users fairly. Now that computing systems are about to be built in which segments of many users' programs are simultaneously present in core and several processors operate simultaneously, another problem arises; scheduling of the necessary swaps and of any requests generated by the various user processes for file memory use. At present, requests are made singly to access file memory; and a request does not occur until the preceding one has been processed. In a system with many simultaneous processes, requests for file memory use will be placed in a queue. This research attempts to find a reasonable model of the way in which user processes demand file memory use and a reasonable and optimal method of handling the queued requests.

It is not possible to construct a detailed model for a computational process in a segmented, multi-programmed system, since no such system exists. However, we can make reasonable, probabilistic assumptions: for example, that segment lengths are Poisson distributed about some mean number of pages; that inter-request times are Poisson distributed; and that a given process will be able to generate requests while its write requests are in service, but will be suspended while any read request is in service.

Having constructed a reasonable model for user processes, we had to consider its request for file usage. The methods of queueing theory were applied, in particular, the following queue disciplines:

1. first-come-first served,
2. shortest-job-first,
3. shortest-access-time-first.

The access time, defined as the rotational positioning delay of a drum or disk file, is measured from the time a channel becomes free until the desired starting location of a request is in position; it is meaningless if there are no requests waiting in the queue. A shortest-access-time-first queue was found to be the most efficient of the three queue disciplines, or any combination of them. Expressions were derived for the average wait in queue and the average number of waiting requests, and the most efficient queue discipline was taken to be the one for which over-all idle time was minimized. Over-all idle time is the idle time of all user processes, plus idle time of all processors, plus idle time of all file memory channels. A method was assumed to exist for optimal core memory allocation, that is, the allocation problem for core is irrelevant to the discussion of file memory scheduling, although we considered allocation schemes. In the analysis, simulation was found to be a most useful tool, because it showed which assumptions were valid and which were not. It also showed the form of probability distributions for waiting times in queue, average number in queue, and so on, that should be expected. Finally, simulation was used to verify the predictions of the mathematical model. (See Denning, Appendix B.)

Input/Output Subsystems - Arthur A. Smith

The execution of input/output instructions in a multi-access computer system may be considered as either simple ordinary instructions, with long execution times when the input/output equipment is part of the central processor, or as a form of communication between the program being executed by the main arithmetic processor and a program written especially for the I/0 subsystem involved, when the I/0 equipment is a separate entity. The former consideration ignores the possibility and the desirability that the central processor execute an entirely separate program while the I/0 is occurring. Execution is implicit in the second viewpoint, which therefore is considered more fruitful.

My studies regard a process as able to force its execution by a particular processor, in particular by a processor suited to the properties of the I/0 device. That is, upon execution of a certain instruction, an image of the state word of the process is transmitted to the I/0 processor, which modifies the state word by the execution of instructions written in its own order code. This I/0 processor, by executing an instruction analogous to the one which started its operation, causes resumption of the usual sequence of instructions, with the appropriately-modified state word. Thus, it executes an instruction which forces the subsequent instructions to be executed by a main (arithmetic) processor. I plan to publish a report on possible realizations of this system and alternatives to it.

Non-Repeatability of Multi-Process Computations - Earl C. Van Horn

A MAC system consists of several processors, a main memory, several file-storage devices, and several input/output devices. The supervisor program of such a system schedules available processors among the various jobs to be done. We have defined the notion of a process as a precise molding of the job to be done. A process is a locus of control or activity within an instruction sequence. We have been studying the relationships among processes and various other system entities and have concentrated on those relationships that are invariant with respect to the way the supervisor schedules available processors among the processes. A collection of processes working together for a single user on a particular problem forms a separate computation. All of the processes of a separate computation may reference the same set of system entities. This set constitutes the sphere of protection associated with the computation.

The possibility exists in a MAC system that different runs of the same computation from the same initial state with the same input data will yield different output data. Such a computation is said to be non-repeatable. Computations running in the present MAC system, CTSS, are repeatable because they always consist of a single process. An attempt to adapt CTSS for multi-process computations would encounter the problem of non-repeatability. Because in practice no program is ever debugged, repeatability cannot be guaranteed to a user unless sufficient conditions for repeatability are enforced by the hardware and supervisor of the system. These sufficient conditions may be published as a set of rules for users to follow in the writing of programs or they may be followed by a compiler in the generation of multi-process programs. In such cases, the burden of establishing repeatability lies on the person debugging the program or compiler, respectively. In order to study the interactions of processes in multi-process computations, we have defined a Multi-Process Automaton of Type A as a model of a multi-process computation. We have postulated one set of conditions sufficient to make a run of such an automaton repeatable. A rigorous proof of this hypothesis is forthcoming. Later, we hope to formulate and prove the necessary conditions for repeatability in such an automaton.

<u>Semantics for Multiprogrammed Computations</u> - Jack B. Dennis and Earl C. Van Horn

The description of computations performed by a multiprogrammed computer system requires functions not available in existing programming languages. In particular, functions related to parallel programming, protection of data and procedure information, and communication among independent computations are usually absent. We have studied the problem posed by these semantic aspects of computations within time-shared computer systems and have proposed a set of meta-instructions through which these essential facilities may be realized. Some of the meta-instructions defined already exist in multiprogrammed computer systems as monitor call operations; however, many of the meta-instructions are yet to be made part of any computer system. The operations performed by meta-instructions are based on the program structure developed in our previous work. A computation is defined by a collection of capabilities represented by entries in a C-list. The process active in a computation may invoke capabilities represented in its C-list, for instance, the addressing of a procedure or data segment or the executing of an I/0 function.

A group of the proposed meta-instructions relates to the control by one computation of other inferior computations. This hierarchical relationship of computations is essential to program debugging within time-sharing systems and to the proper handling of faults or exceptional conditions encountered by a computation process. Other meta-instructions relate to the accessing and control of a directory structure for retained objects, principally segments of procedure and data. We suggest a hierarchical directory structure associated with each intelligence that uses the computer system in order to facilitate the choice of unique identifiers for retained objects and make the sharing of objects among computations feasible.

Optimal Allocation of System Resources - Robert L. Potter

The problem of optimal memory and processor allocation in a multiprocessor, time-shared computer is being studied. It is formulated as a two-space mapping problem with emphasis on the use of probabilistic methods. A computer has a certain number of memory and of processor units available. A two-dimensional space can be visualized. I constructed two spaces with the same normal axes, a memory unit axis and a processor unit axis; one is called "resource" space, the other, "needs" space. The resource needs of users are distributed in needs space. An allocation algorithm assigns each user to a portion of resource space. This is a mapping from needs space onto resource space, and the mapping function must assign resources equitably and efficiently.

The probability densities of memory and processor needs per user will not be known until measurements can be made on an appropriate system. We expect the densities, which are time-dependent, to have the same general appearance; they are zero at zero need, rise to some maximum, and then fall off toward zero as the need becomes infinite.

The mapping function is not easy to determine; it must include cost, efficiency, and user-satisfaction factors reflected in the way distribution in needs spaces adapts to the system. Cost factors may be defined in many ways: as the allocation scheme which minimizes cost, in one case; or wasted time or resource units in another. User-satisfaction factors are elusive of definition until a required system is built.

The mapping must be time-dependent, because the present MAC system considers allocation primarily in terms of program length, which discriminates against long programs. We need a more sophisticated approach in which the running status of a user depends not only on program length, but also on the nature of the resources employed since the beginning of computation. Thus, a mapping function must consider a user's position in both resource space and needs space for the previous few moments and the present moment. (See Potter, Appendix B.)

Automatic Flowcharting - Daniel U. Wilde

The objective of this research is development of a tool to aid advanced programmers in analyzing and debuggin digital computer programs written for general-purpose, single-address machines. Input to the analysis program is a BCD output-listing tape from the 7094 FAP assembler. Output from the analysis program will be a flow chart in symbolic form showing the effects of control flow and data flow. The following development work has been completed:

A. DATA GATHERING

On the first pass, the analysis program reads a BCD assembly tape and produces the following tables as a function of the instruction location:

a. Entry and Exit transfer tables,
b. Active and Passive reference tables,

c. Data and Storage pseudo-operation tables,
d. A Symbol table,
e. An Entry-Point or Starting-Location table, and
f. A First-Instruction and Last-Instruction location table.

B. DATA REDUCTION

The second pass of the analysis program processes the tables constructed during the first pass. Examples of the processing are:

a. The Entry and Exit tables are used to divide the program into "blocks". A "block" is a set of instructions between a transfer entry point and the next transfer exit point. Thus, a "block" is completely processed if its first member is executed.

b. The Active and Passive reference tables are used to construct the Constant and the Result tables.

c. The Data and Storage tables are used, along with the Constant and Result tables, to determine if the input program changes or modifies any of its own instructions.

C. MICRO DATA COMPRESSION

The third pass of the analysis program processes the tables of the second pass. Examples of the processing are:

a. The Block table is used to determine if all non-data blocks can be reached from either the starting location or the various entry locations. If a block cannot be reached, then the topology of the surrounding blocks is studied to determine what connection assumptions should be made.

b. The connected Block table and the Active and Passive tables are then used to construct the Latest reference table as a function of program topology for each passive reference.

D. MACRO DATA COMPRESSION

a. Each active reference represents data changes that are made by the program being analyzed. By using the Latest table entry for each Passive table entry, a symbolic expression for each Active table entry can be constructed as a function of program topology.

b. Cross reference information showing Latest values of used registers, e.g., index registers, is also provided.

The final flow-chart output can be set to show either a topology-only level or topology-and-data-processing level. Further refinement of these output formats is necessary, and this requires further consideration of possible data-flow representations.

Visual Information Processing - Herbert M. Teager

Major effort, under ONR Contract Nonr-1841(69) and JSEP Contract DA36-039-AMC-03200(E), has been devoted to the area of human sensory processing, primarily visual. The object of this effort has been three-fold. First, from a pragmatic viewpoint, to provide for handwritten and drawn symbolic input to digital systems with a latitude and scope approaching normal human practice. Second, to develop a testable theory of the visual processing that occurs in living systems, since our theories with respect to pattern recognition are generally inadequate for almost any of the commonly encountered facets of human visual capability, such as visual memory, depth perception, face and object recognition, et al. Third, to understand the "what" and "how" with respect to sensory processing, which is a fascinating and potentially fruitful field of scientific research in its own right.

The work that was accomplished has been presented as a paper, entitled Multidimensional Visual Information Processing, to the New York Academy of Sciences. There is not the space available to duplicate the material of that paper here. However, the results with respect to development of a useful system are summarized below.

It is possible to build a simple hardware system to accomplish real-time character recognition, when connected to a "Teager table/Rand tablet", for a known individual's writing with the following constraints and capabilities: Printed-writing script must be disconnected and at the general level of complexity of Roman character fonts, with no limitation on writing speed. The system is invariant to size, location, rotation skew, and all other affine transformations generated by oblique projection. Within the limits of character complexity and delay-line storage capacity, the system is unlimited with respect to character font. In practice, a font of no more than 256 different characters is optimal with respect to the hardware. These conclusions have been verified, with simulated hardware, by an extended period of experimentation on writing from a varied group of subjects, during which the very low error rate that occurred (less than 3%) could be traced almost exclusively to the use of symbols that were completely new to the system.

Further work is being expended to build the hardware, as well as extend its capability to non-real-time recognition, unknown source, connected writing, and a character font at the level of complexity of Chinese and shorthand.

With regard to the development of a tentative theory for visual processing of two-dimensional information, conclusions have been reached with respect to the form of measurement and processing which may well be occurring in living systems if the known experimental, psychological, and artistic evidence of line, shape, and form incoding is considered. These measurements, in fact, form a basis for the processing system that has been developed. Evidence for such measurements and analogous ones in other sensory modalities are being sought, and consideration is being given to means of artificially accomplishing such processing with contemporary processing and sensing hardware.

One tentative theoretical conclusion with respect to visual processing is that optic nerve connections are far too few in number to be carrying information to the brain in the assumed one-way, binary fashion. Obviously, analogous speculations are for future exploration in conjunction with appropriate life and medical scientists.

At the operational level, a multiplexing system of hard-copy graphical outputs and its attendent programming was turned over to the ESL group; and Eric Westerfeld developed a program to translate hand-drawn flow charts into machine-compilable MAD programs. (See Westerfeld, Appendix B)

COMPUTER OPERATION

CTSS Operation and Equipment

Non-Academic Research Staff

M. V. Solomita

Operating Staff

G. Andrews
D. Borsini
R. Brenner
K. Carley
R. Cerrato
G. Dardis
T. Dattilo
R. A. Degan
R. G. Hart
R. McNamara
R. Moore
S. Nelson
G. Noseworthy
M. Pagliarulo
R. Parker
E. Reardon
J. W. Waclawski

CTSS Operation and Equipment - M. V. Solomita

During this reporting period the IBM 7094 was scheduled for operation twenty four hours a day, seven days a week with the exception of a shut-down for the Christmas and New Years Holidays. The computer was also not in operation during the installation of the IBM 7289 channel and 7320A high speed drums in September, and the replacing of the IBM 1301 Disk storage file with the IBM 2302 Disk storage file in January. These changes have helped increase the speed and storage capacity of the Compatible Time-Sharing System.

In June of 1965 the GE Model 635 began arriving and by June 15th the installation of the equipment began. We expect operation of the computer to start during the third quarter of 1965.

The IBM 7094 Data Processing System configuration now consists of the following:

	Type	Description
	7109	Control Processing Unit
	7110	Control Processing Unit
2	7302	Magnetic Core Storage (32, 768 words each)
	7151	Console Control Unit
3	7607	Data Channels
3	7617	Data Channel Consoles
2	7909	Data Channels
	7606	Multiplexor
	7608	Power Converter
	7618	Power Control
	7631	File Control Unit
	2302	Disk Storage Unit (234 million characters)
	7320	Drum Storage Unit (1, 118, 400 characters)
	7750	Data Communication Channel
2	7320A	Drum Storage Units (1, 118, 400 characters each)
12	729VI	Magnetic Tape Units
	716	Alphabetic Printer
	711	Punched Card Reader
	721	Card Punch
	7289	Data Channel

Peripheral (Off-Line) Equipment:

	Type	
	1401	Central Processing Unit (4, 000 characters)
	1402	Card Read-Punch
2	1402	Printers (132 character/line each)
4	729-V	Magnetic Tape Units

COMPUTER SYSTEM RESEARCH

Research on the Compatible Time-Sharing System

MULTICS' Hardware System Design

Academic Staff

F. J. Corbató
E. L. Glaser
J. W. Poduska
J. H. Saltzer

Non-Academic Research Staff

M. J. Bailey
R. U. Bayles
M. E. Child
R. Creasy
P. A. Crisman
C. B. Cushing
R. C. Daley
S. D. Dunten
D. J. Edwards
R. Fenichel
R. M. Graham
T. Hastings
S. Kidd
E. Kliman
D. E. Oppert
R. Orenstein
L. Pouzin
G. C. Schroeder
L. Varian
M. R. Wagner

Research Assistants and other Students

D. R. Widrig, Jr.
J. M. Winett

Research on the Compatible Time-Sharing System - Fernando J. Corbató

The system programming group has continued developing the Compatible Time-Sharing System (CTSS) for the IBM 7094 computer of Project MAC. This evolution has allowed the constant modification and testing of new ideas in a framework of man-machine interaction. In fact, the ability of the system to evolve with time has been a basic design goal. Hardware requirements in the past year have also caused further programming requirements. Two IBM 7320A high-speed drums, attached to the MAC computer in August, have significantly improved response times.

Thus, time-sharing at Project MAC has been: 1) a service facility which has allowed individual users to explore the consequences of man-machine interaction, and 2) a laboratory in which system programmers can develop the tools needed for effective man-machine interaction. The 7094 system is still in the process of being tuned, though it is incapable of being tuned to any high degree because the basic multi-programming ability and storage allocation techniques are very difficult. For this reason, a second-generation time-sharing system called MULTICS (for Multiplexed Information and Computing Service) will be implemented initially on a GE 645 as a development project with the Bell Telephone Laboratories and the General Electric Computer Department.

The Multics Project was started this year, and papers describing the system are to be presented at the Fall Joint Computer Conference, 1965. The major design decisions which have been made are in the area of hardware. A separate report by E. L. Glaser covers hardware design. Software plans have been harder to firmly establish.

Tuning and improvement of the 7094 system has been done with two motives; there has been the desire to develop an accurate pilot model of possible future systems, and there has been great interest to learn from mistakes. On both counts it is felt that considerable success has been achieved during the past year. A detailed description of the 7094 system may be found in the Compatible Time-Sharing System, 2nd edition, M.I.T. Press, 1965. What follows is a brief review of the major elements of the evolvement of the 7094 system.

It has been recognized that the disk file system is a major and central part of a time-sharing system. Already CTSS has one of the most elaborate file systems developed. Nevertheless, it is also recognized that the first version of the file system is inadequate for a more highly-tuned time-sharing system. Some of the changes that have been improvements are the salvager and an improved input/output editor for placing or removing material from the disk file. These changes have improved the reliability and smoothness of system operation inasmuch as the features involved occupy pivotal software roles and are expected to always function without mishap.

However, of major importance has been the design of the second-generation file system. This major redesign is expected to correct four areas of difficulty. One, it should be possible to share files among many users, thereby allowing multiple reading and suitable interlocking upon writing. Two, multiprogramming should be possible such that more than one user can be using the file system simultaneously. Three, it should be possible to use magnetic tapes from within the time-sharing system. Four, it should be possible to incrementally dump only newly-created material, on a continuous basis, rather than have a massive dump periodically (e. g., once a day) of the entire contents of the disk. This process is being developed under the program name DAEMON. Basic design was concluded in the fall of 1964. Implementation was begun and is now approaching the terminal stages. The implementation of this new file system is considered of prime importance, as it gives the programming staff an opportunity to shake-down the multitude of ideas which are involved in anticipation of MULTICS.

Other improvements to CTSS have been the introduction of a class of new editing programs for typing in programs, text, etc. Of notable interest has been the development of the TYPSET and RUNOFF commands for English text and the corresponding ED and EDL commands for program text. These programs are based on a synthesis of many editing techniques by J. Saltzer and have mostly replaced the need for the previous editors such as MEMO, MODIFY, DITTO, INPUT, EDIT, and FILE. The new editors are distinguished by a high degree of sophistication for man-machine interaction as well as context editing techniques which avoid fixed format requirements.

In addition, nearly all commands of the system have had some corrections or improvements. Particularly notable are: inclusion of a new version of the MAD compiler; cleaning up and fixing up of the FAP command to allow use of more compacted input files; improvement of the LOAD command to allow three different program libraries and thus speed up library searching; and reprogramming of FILE, RUNCOM, SPLIT, COMBIN, SNOBAL and other commands. Further new features have been added in the form of interconsole message facilities and the ability to "attach" other consoles. In addition, new debugging techniques have been introduced, the most notable being the MADBUG program of R. Fabry for the FAPDBG-like debugging of MADBUG programs. The MADBUG program is considered only a prototype of future debugging techniques, as the present CTSS implementation forces it to be relatively clumsy. Nevertheless, the design goal, that the MAD language user need not know machine language, has largely been met.

Finally, there have been numerous attempts to find solutions for the difficult problem area of system documentation. There have been many attempts to compact disc space requirements and allow larger utilization of the available secondary storage capacity. In particular, the ARCHIVE and CRUNCH commands have been developed. In addition, the LOG command has been developed to notify system users and allow user teams to keep posted all system changes of current interest. Moreover, the system programmers themselves have developed conventions and procedures in an attempt to systematize record keeping for a large system. In addition, facilities have been developed for semi-automatic "carry" of programs back and forth between the two 7094 computers to allow smoother and easier maintenance of two slightly different CTSS systems.

Of major importance in the orderly development of the new system has been a complete revision of the present CTSS Programmer's Guide to include information from many CTSS Bulletins and important MAC memos describing changes in the features of the system. This revised manual will be maintained on-line in the computer and is available to any user at any hour of the day or night whether he be 20 feet or 2000 miles away from the Computer. In this way, users may keep abreast of the latest system changes. In particular, a user is given the ability to read a table of contents of the changes listed in reverse chronological order. It is expected that the manual, which can be maintained by the system programming staff in as dynamic a state as the system itself, will be a significant part of the answer to documentation difficulties of future remote-console systems.

MULTICS Hardware System Design - Edward L. Glaser

During the reporting period, the hardware design for the new Multiplexed Information and Computing Service (MULTICS) time-sharing system was formulated and frozen. This design consists of rather extensive modifications to a General Electric 635 computer. The resulting computing system is to be known as the General Electric 645 computer. Extensive modifications were carried into a number of different areas. First of all, the addressing logic was enhanced to incorporate the concepts of segmentation and paging.* Additionally, some new instructions were added to the instruction repertoire, and the interrupt logic was enhanced, making it possible to interrupt an instruction part way through execution, and subsequently continue with the execution of this instruction. These explicit modifications will enable the system to handle software envisioned for the new time-sharing system.

The second major area of design was that of a general I/O control unit. This I/O control unit incorporated functions of both a standard I/O control channel and a communications computer. Thus, the same equivalent logic could be used to communicate with both standard computer peripherals, such as magnetic tape, disc, and card equipment, and with varying communication lines from teletype speed to very high speed for displays.

---

*See (1) Dennis, J. B., "Program Structure in a Multi-Access Computer," Project MAC Technical Report MAC-TR-11; and (2) Dennis, J. B. and Glaser, E. L., "The Structure of On-Line Information Processing Systems," Information System Sciences, Proceedings of the Second Congress, Spartan Books, Inc.

The generality of design is intended to make modular software in the new time-sharing system not only feasible but efficient. Further, specific features are incorporated into the design of the I/O equipment control unit to facilitate processing of many interrupts in a short period of time, which is necessary on any system connected to a large number of remote terminals.

The third major area of design is that of a new high-speed drum controller to be used in conjunction with a large-capacity drum. The availability of this new drum controller with its associated drum will facilitate time-sharing for two reasons: first, the transfer rate between the drum and core memory is approximately fifty percent of memory speed; second, because of the way the drum controller operates, it is possible on a statistical basis to ignore any latency due to the rotary access to the drum.

Details on all of these design areas will be available in papers to be presented at the Fall Joint Computer Conference, 1965.

BLANK PAGE

ELECTRONIC SYSTEMS LABORATORY

Introduction

Display Systems Research

Computer-Aided Design

Computer-Aided Electronic Circuit Design

Accelerometer System Studies

Simulation of Strapped-Down Navigation Systems

Academic Staff

J. F. Reintjes
M. L. Dertouzos
A. K. Susskind

Non-Academic Research Staff

R. J. Bigelow
C. G. Feldmann
U. F. Gronemann
D. R. Haring
F. B. Hills
J. Katzenelson
P. T. Ladd
R. O. Ladson
C. A. Lang
R. B. Lapin
J. A. Parisot
R. B. Polansky
D. T. Ross
R. H. Stotz
D. E. Thornhill
J. F. Walsh
J. E. Ward

C. W. Bower - Boeing Company
R. K Coe - United Aircraft Corp.
B. T. Fox - Sandia Corp.
L. H. Haines - I.B.M. Corp.
J. R. Kennedy - Lockheed-Georgia Co.
J. Martyniak - North American Aviation, Inc.
H. W. Spencer - Grumman Aircraft Engineering Co.

Research Assistants and other Students

T. B. Cheek
A. A. Dvorak, Jr.
D. S. Evans
Z. Fluhr
E. G. Guttmann
G. C. Ling
A. Malhotra
K. Olshansky
J. E. Rodriguez
P. J. Santos, Jr.
W. D. Stratton
C. W Therrien
B. L. Wolman

Introduction - J. Francis Reintjes

Several research groups of the Electronic Systems Laboratory are participating in Project MAC. Display systems research has been conducted in support of the Laboratory effort on computer-aided design and for the purpose of enhancing the output-display capabilities of the MAC time-computer system.

The Computer Applications Group continued its research in fundamentals of man-machine problem-solving, with emphasis upon the development of generalized computing techniques which are expected to be applicable to a wide variety of engineering design problems. Emphasis in this project is on the use of computers as an aid in mechanical design.

Computer-aided electronic circuit design was also investigated under a grant from the National Aeronautics and Space Administration. The main efforts of the group were devoted to: analysis techniques for linear and nonlinear networks, logic design and synthesis through use of the MAC computer, design of threshold logic, and the development of a low-cost teletype-operated graphical display.

The Project MAC time-sharing system has also been very useful for investigating simulation of force-rebalance accelerometer output data. The objective of this work is to obtain maximum output information by using specialized data processing. In another aspect of the same research program, the group investigating computations peculiar to strapped-down navigation systems found the Project MAC PDP-6 to be highly useful.

A total of 16 staff members and 12 students in the Electronic Systems Laboratory have benefited in their research through the use of Project MAC facilities.

Display Systems Research - John E. Ward and Robert H. Stotz

A. ESL DISPLAY CONSOLE

The ESL Display Console was designed to satisfy needs of the M.I.T. Computer-Aided Design Project, and its construction was supported by the Air Force Materials Laboratory, Wright-Patterson Air Force Base, under Contract AF-33(657)-10954. The unusual features of this display system, which have permitted its routine use in the Project MAC time-sharing system since January, 1964, are: its ability to time share the 7094 memory through the Direct Data Channel; its incremental digital line generation, scaling, and rotation system; and its automatic-hardware pen-tracking system. During the past year, the system capabilities have been extended, with Project MAC support, by adding a second operator station (consisting of a slave display scope and associated manual inputs) which time-shares the display generation electronics, thus allowing two operators to work simultaneously on different problems. Typical current usages, which are reported in other sections of this progress report, include: three-dimensional drawing programs, parametric surface studies, and portrayal of stress contours by the Computer-Aided Design Project; three-dimensional models of protein molecules by the Biology Department; modeling of the human vocal tract by the RLE Speech Group; design of highway interchanges by the Civil Engineering Department; and general graphical output by a number of other groups who use the display in an interactive way. Usage currently averages about 16 hours per day for each of the two console stations.

The M.I.T. Computation Center has recently decided to install a complete duplicate of the ESL Console on its time-shared 7094 to extend the availability of display facilities to a larger segment of the M.I.T. community. The systems will be compatible and Project MAC display software can be used directly. Digital Equipment Corporation is building the electronic portions of this second system from M.I.T. drawings, and the manual input devices are being fabricated in the Electronic Systems Laboratory. This second console, which is supported by an NSF Grant, is scheduled for installation in November, 1965.

B. STORAGE-TUBE DISPLAYS

Experiments with driving "one-shot" storage tube displays from analog deflection voltages of the ESL Display Console have been quite successful, although the quality of the present storage tubes leaves something to be desired. Two such displays, connected to the ESL console by three coaxial cables, have been in experimental use during the past few months. Programming and control circuit modifications to permit the addressing of up to ten remote storage scopes for transmission of "one shot" pictures have been accomplished. Figure 3 shows typical storage displays. These require only about one half second of console time to create, but are retained for periods up to one hour, unless erased manually or by the computer. Experiments are continuing with the use of analog data sets on three telephone circuits (for X and Y deflection voltages and intensification pulses) for remote operation of storage scopes in this mode at greater distances. Noise and linearity characteristics of the telephone circuits appear satisfactory, though the difference in time delays on the three circuits is proving troublesome.

C. DISPLAYS FOR THE GE 645 COMPUTER

Study of future display requirements for the Project MAC GE 645 Computer has resulted in identification of two goals: 1) a modest number (10-15) of remote sophisticated displays, with capabilities similar to the present ESL Console; and 2) a large number (perhaps 500) of low-cost, limited-capability displays which can be located at every MAC terminal.

Planning for low-cost remote display scopes currently centers around a digitally-controlled unit (rather than the analog transmission system discussed in B, previously), with line and character generation being performed locally. A set of desirable characteristics for a unit which could operate from standard 2000-bit-per-second Dataphones has been prepared and is being discussed with the manufacturers of display equipment. Also, research work is being conducted on circuits for use in such a device. One master's thesis project by Mr. T. B. Cheek concerns new approaches to character generation, with the objective of exploiting low-speed requirements to achieve very low cost. A low-cost, hybrid digital/analog, line-generation system is also being designed to operate from a Dataphone. This approach combines a binary rate multiplier with an analog integrator, thus

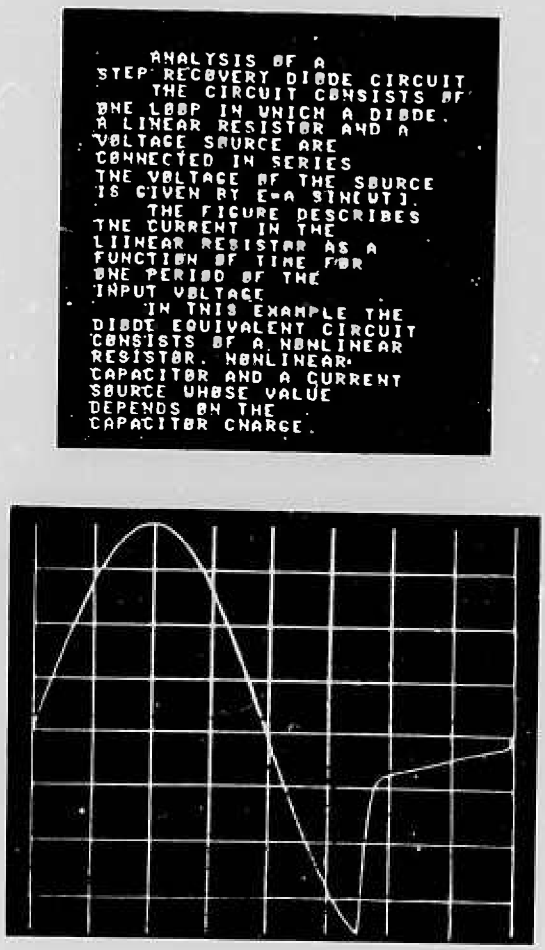


Figure 3. Sample Storage Tube Displays

eliminating the need for digital-to-analog converters. A prototype unit, with parts costing about \$1,000, should be ready for initial tests by September, 1965.

Operating more sophisticated remote displays (like the ESL Console) directly from the central computer (in the manner of the present ESL Console) will present definite communications problems, unless a flexible, low-cost, multi-megacycle, two-way link becomes available. Also, the load on the central computer must be considered. Our experience indicates that the best approach is to use a small satellite computer with each display (or group of displays) to buffer display data transmitted over medium-speed telephone circuits (4 to 10×10^4 bits per second). The satellite computer would also perform real-time display processing, such as pen tracking, rotation calculations, and interrupt analysis. One such computer-buffered display (the DEC 338) will be available at Project MAC during the coming year for experimental evaluation and study of programming problems. (Note that the small computer program can be considered part of the supervisor program of the central machine.)

D. TV SCAN CONVERSION

The display group has completed the design for an image-maintaining display system based on a TV scan converter for the U. S. Naval Underwater Ordnance Station under Contract N140(122)-76148B. Computer-generated pictures are drawn on the storage surface of a scan-conversion tube by slaving its deflection inputs to an existing DEC Type 340 display system. Then, by switching the deflection inputs to a TV raster, the stored picture may be viewed for periods up to five minutes on standard TV monitors. A 1203-line scan is used to provide high resolution. Selective erasure or writing of up to three characters will be possible during each TV vertical retrace time, so that stored pictures may be altered and updated without requiring a complete regeneration by the computer. This system will be experimentally tested on Project MAC's PDP-6 computer before being delivered to the Navy later in the year.

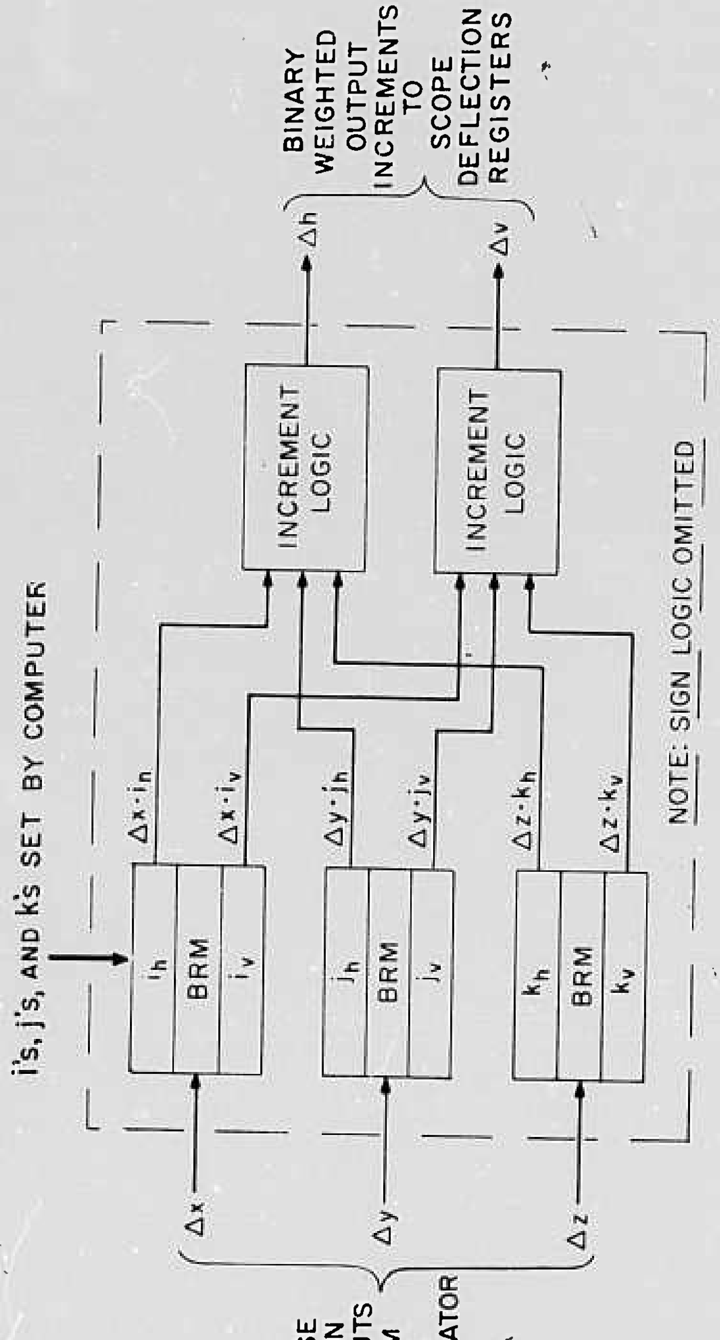


Figure 4. Present BRM Rotation Matrix of the ESL Console

E. IMPROVED DISPLAY TECHNOLOGY

A number of other aspects of display technology are being studied as thesis projects. Mr. W. D. Stratton is working on a new high-speed pen-tracking system, based on the amplitude comparison of light-pen responses to displayed points in a tracking pattern, similar to radar conical-tracking schemes. By replacing the present digital point-counting schemes, it is hoped that pen-tracking time can be reduced by a factor of perhaps 20 to 1. The new electrostatic "beam pen" being developed by the group is being used in this work, although the technique could also be applied to light pens.

The possibilities of infrared (IR) data links for remote operation of displays is being investigated in a thesis project by Mr. George Ling. We have obtained an experimental IR link, on loan from the Lincoln Laboratory, that is capable of 10^7 bits per second over a two-mile range, and have evaluated its performance in transmitting digitally-coded messages. The link is now being set up to transmit display information from the ESL Console in Technology Square to a storage scope in the Electronic Systems Laboratory, a distance of about 600 yards.

As part of the continuing program to improve the ESL Display Console, Mr. Endre Guttman completed his thesis study of a digital differential-analyzer (DDA) rotation matrix to replace the present binary rate-multiplier (BRM) rotation matrix. In the BRM matrix, shown in Figure 4, the 10-bit signed numbers i_h, i_v, etc., are set by the computer and are normally the vector cosines specifying rotation between the x, y, z and h, v, d coordinate systems (depth coordinate, d, is not computed). Scaling of these numbers controls the size of the displayed image. The boxes labeled "increment logic" combine and buffer the BRM outputs, releasing them to the scope deflection registers in binary weights of 1, 2, 4, or 8 steps, which the registers (up-down counters) can accept.

Although the BRM matrix is generally satisfactory, it has significant round-off errors which create picture distortions. Also, the buffering action of the increment logic adds some roughness to displayed lines.

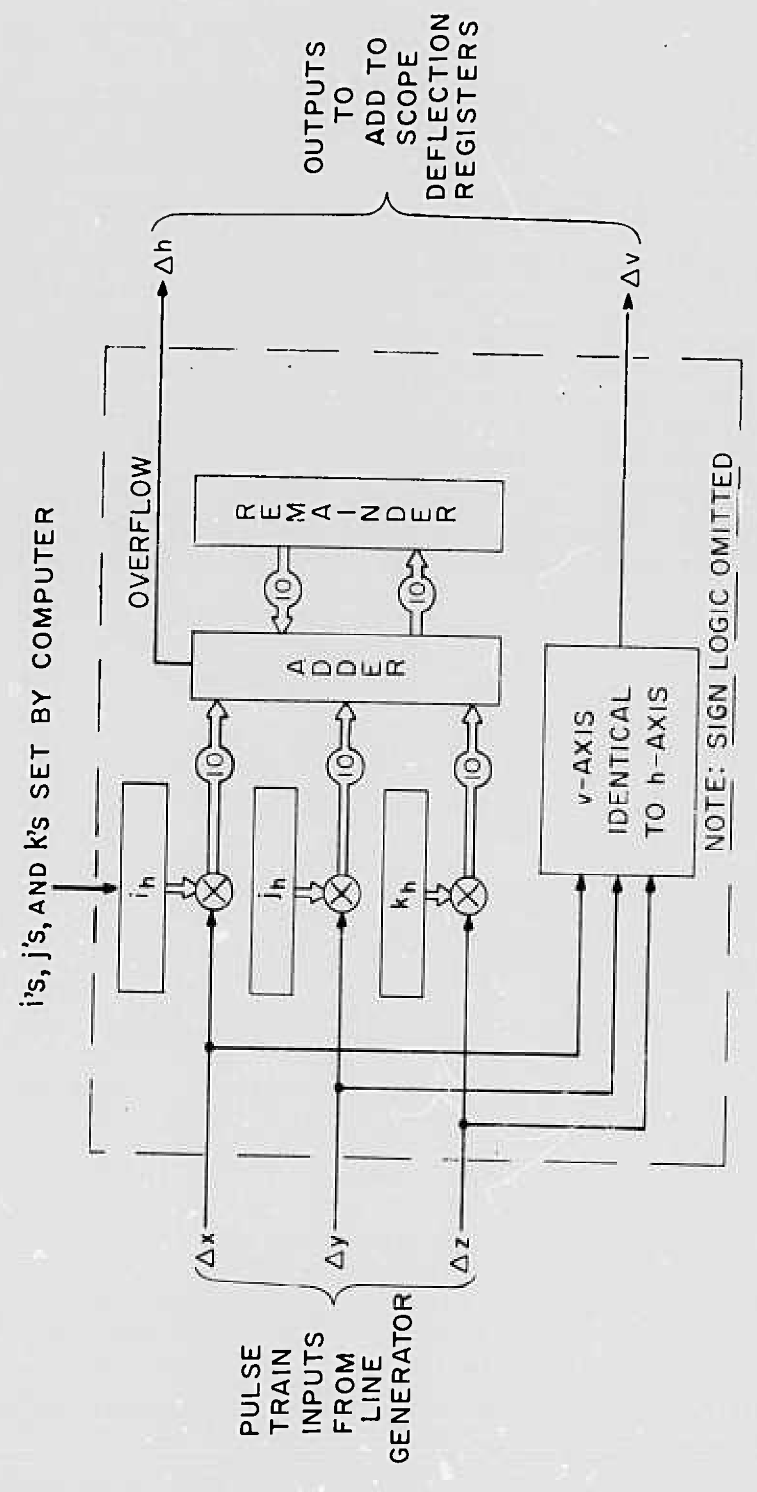


Figure 5. Proposed DDA Rotation Matrix

Figure 5 shows the proposed matrix, which would have two identical three-input parallel DDA's, one for h, and one for v. The advantage of the DDA over the BRM is that it has a remainder register to "remember" round-off in a particular step of the calculation, and apply it to the subsequent step. Also, the h and v outputs would be obtained directly, eliminating part of the function of the increment logic. The increment logic could then be eliminated completely by converting the first few stages of the deflection registers from counters to adders.

A DDA for this purpose, however, must be able to add four, ten-bit, signed numbers in a total elapsed time of only 1.5 microsecond. It was shown that multi-input threshold-logic elements offer the only feasible way of meeting such a severe speed requirement at reasonable cost. A potentially suitable six-input threshold-adder stage (four register inputs, plus two carries) requiring only three transistors was designed and breadboarded. It was found to have a transition time of 80 nanoseconds; fast enough for this purpose. Further development and testing of threshold circuits will be necessary, however, before the DDA rotation matrix can be constructed.

F. FUTURE PLANS

All reported activities will continue during the coming year, with particular emphasis on developing the required display technology for future graphical man-machine communication and studying the hardware/software problems in interfacing large numbers of displays with the next generation time-sharing computers.

Computer-Aided Design - Douglas T. Ross and Clarence G. Feldmann

The Computer Applications Group, along with faculty and staff of the Mechanical Engineering Department, has continued its Computer-Aided Design work, with emphasis on problems of mechanical design. This program is being sponsored by the Air Force Materials Laboratory, Wright-Patterson Air Force Base, under Contract AF-33(657)-10954. The work is an evolution of earlier work, conducted at ESL, to develop the Automatically Programmed Tool (APT) System now being used by industry to program numerically-controlled machine tools. The approach being taken to computer-aided design is sufficiently general to make it applicable to a wide variety of design problems in engineering and science.

The current research effort has four phases. The first phase involves research in fundamentals of man-machine problem-solving, the structure of language and language processing, the modeling of problems, and the development of generalized computing techniques. The second phase concerns application of the theoretical foundations resulting from this research to the design and construction of a generalized man-machine problem-solving system, including development of a powerful language for writing, operating, and debugging programs, and facilities for efficient construction of specialized problem-oriented systems. The third phase is actual development of such systems for specific design areas, and the remaining phase is adaptation and application of special systems by designers to meet their individual needs.

In the realm of generalized languages and systems, the AED-0 system (standing for Automated Engineering Design) was the first system completed by the group. It has been in operation for over a year and is being employed extensively by members of the Design Group and by other groups at Project MAC. More recently, through the efforts of Mr. Feldmann and his associates, AED-0 has been made available as a batch-processing system and distributed to interested industrial users. Mr. Ross presented a series of lectures on the salient features of AED-0 language, and his notes now constitute the basic AED-0 Programmers' Guide.

During the year, many substantial additions were made to AED-0 in order to increase its power for writing advanced programming systems. The language is based on Algol-60, with many additions and modifications to permit construction and manipulation of complex data structures, including the ability to reference subparts of computer words for efficient use of storage. The system also includes an elaborate Macro Preprocessor which permits extensive manipulation of the input string of characters before compilation, thus providing the user with greatly enhanced expressive capabilities. A further feature is that the Preprocessor includes facilities for regenerating AED-0 language programs in a meaningful format, which exhibits the logical structure of the program. Combined with flexible "comment" and "remark" features, this capability makes AED-0 programs almost self-documenting, minimizing the need for elaborate flow diagrams to describe programs. A source-language debugging system has been written, so that the user may trouble-shoot programs at the AED-0 language level, with little concern for the actual machine code generated by the compiler. AED-0 development is now complete, and primary attention is now focused on its use as a tool to achieve even more sophisticated generalized programming and compilation systems.

The immediate successor to AED-0 will be the AED-1 System now under development. A salient feature of AED-1 is that it will be highly machine-independent. That is, it is being designed so that both the language and techniques used to compile machine-code programs will be compatible with virtually any type of large-scale computer. Several valuable characteristics of AED-0 will be contained in its successor, as well as new features to give added flexibility and capability. The AED-1 System will be able to process AED-0 as well as AED-1 language programs. Several building-block (subsystem) programs are being written to achieve this goal, including a subsystem (AED Jr.) which permits the user to define and check out descriptions of arbitrary programming languages.

With respect to specialized, problem-oriented programming systems, the group's first major system in this category is now being prepared. Called CADET (Computer-Aided Design Experimental Translator), it permits users to carry out computer-aided design applications using verbal or graphical language or both. That is, its purpose is to enable construction and manipulation of graphical and analytic structures. CADET will also permit analytic functions and physical properties to be associated with graphical representations for things. Since programs defining behavioral properties of graphical elements may take arbitrary form, they are written in standard AED language.

To assist the initial development of CADET, which will be a general-purpose problem-oriented system, several specific design problems are being explored through its use. One of these is concerned with the general problem of three-dimensional shape description, and will use the generalized parametric surface patches developed by Professor Coons of the Mechanical Engineering Department, as well as the three-dimensional pseudo-pen program written by Mr. Polansky, and the routines prepared by Mr. Lang for ESL's Graphical Display Console. Another application is in the area of linear and nonlinear electronic circuits, described in the next section.

A further aspect of the entire program in Computer-Aided Design is the work being performed on cathode-ray-tube displays for graphical manipulation of design information and symbols. This work was described in the previous section.

An important goal of the Computer-Aided Design Project is the dissemination of information generated by the group to interested users outside M.I.T. It was our pleasure to have seven system programmers from industry as guests of ESL and Project MAC for the past year. Our guests have participated in developing the programming systems described previously, and they plan to apply and further expand these basic concepts upon return to their respective organizations. This program of cooperative research and development with visiting staff from industry will continue as an important aspect of ESL's work in computer-aided design.

Computer-Aided Electronic Circuit Design - Michael L. Dertouzos and J. Francis Reintjes

A. SIMULATION OF ELECTRICAL NETWORKS

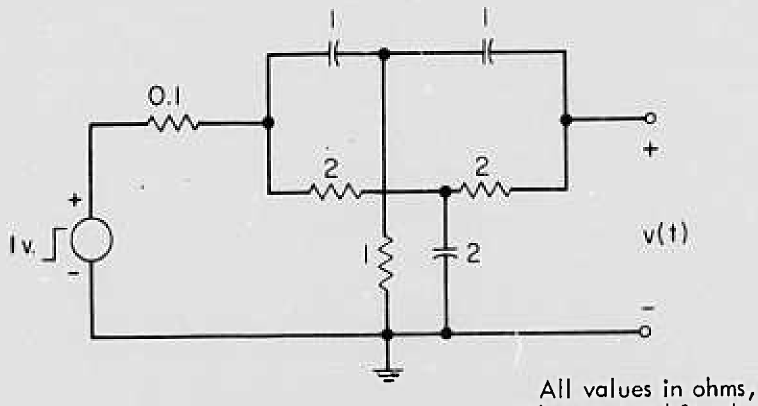
A method is being investigated for on-line simulation of electrical networks. This method treats networks consisting of linear and nonlinear resistors, as well as inductors and capacitors. A given network can be excited with voltage and current sources which may be arbitrary functions of time. The voltage response across any node in the network may then be called for as output.

The computer model for the network consists of lists of storage registers representing branches connected to other lists representing nodes. The method for solution at each instant of time is as follows: The nodes are assigned fixed node potentials. The voltage across each branch, and hence the current through it, is determined by the two nodes to which it is connected. Equilibrium requires the sum of the branch currents at each node to be zero. If the sum is not zero, new node potentials closer to the equilibrium values are computed by perturbing the old ones in a prescribed manner. This process is repeated until the equilibrium values are obtained within a specified tolerance.

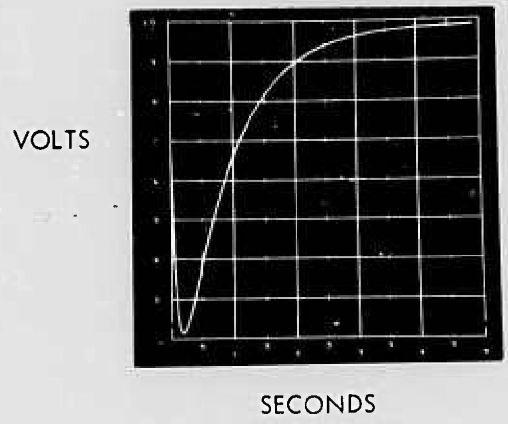
Thus far, a computer program for use on the Project MAC time-sharing system has been written to apply the foregoing method to linear RLC networks excited with step and sinusoidal current and voltage sources. The program computes an accurate solution to any given linear network. Computer time for solution of networks containing up to about 8 nodes and 12 branches is on the order of a few seconds. An example of a circuit analyzed by the foregoing program is shown in Figure 6.

B. DISPLAY PROGRAM FOR ELECTRONIC CIRCUITS

A computer program has been written which allows an operator to quickly and easily portray planar electronic circuits on the Electronic Systems Laboratory Console Display at Project MAC. Circuit elements are formed by moving a tracking light pen to the desired position on the screen and actuating an "element button." Circuit characteristics are described in a list structure with pointers. Used in conjunction with the analysis program



(a) Twin-T Filter



(b) voltage v(t) in Response to Unit Step

Figure 6. Response of a Twin-T Filter as Computed and Plotted by the Circuit-Analysis Program

described previously, circuits with multiple elements can be displayed and their response to various types of excitation readily determined.

C. ON-LINE SWITCHING-FUNCTION SYNTHESIS

The central aim of logical design is the synthesis of any given switching function, in terms of given sets of elementary building blocks, for the optimization of some performance index in the presence of constraints. Although the present state-of-the-art yields algorithmic methods for the solution of certain specific instances of the problem (such as minimization of building-block inputs with a two-level, AND-OR realization), no fully-algorithmic method exists for the solution of the more general problem.

Solutions to the general problem are being developed through use of an on-line process, where the machine accomplishes those computational tasks which can be algorithmically specified, and the user provides those decisions which he is better qualified to make. The machine portion of the system is based on a set of heuristic procedures which, subject to certain conditions, guarantee convergence of the process. Results obtained to date indicate that this method of synthesis yields more economical structures than those using successive local-optimization procedures, and does not depend on impractical (and usually impossible) exhaustive searches through all possible solutions. It is expected that a report on this work will be issued during the Fall, 1965.

D. THRESHOLD ELEMENT REALIZABILITY

Previous work has shown that the problem of determining whether an arbitrary Boolean function of N variables is realizable with a single threshold element can be reduced to the problem of minimizing a quantity which is a function of N+1 variables. Several properties of this quantity are being derived and studied. It can be shown, for example, that the quantity may be interpreted geometrically as a structure of intersecting N+1 dimensional hyperplanes. Each of these hyperplanes corresponds to some N-variable Boolean function, and if the given Boolean function is realizable, its associated hyperplane will be "flat" with zero height. The hyperplanes are so situated that they form the boundary of a convex body and hence the quantity contains no "pockets" of local minima. This property enables a computer to employ hill-descending techniques in N+1 space which seek a local minimum point. Since, by virtue of the convex nature of the quantity, any point satisfying local minimum conditions must also satisfy absolute minimum conditions, hill-descending techniques will lead to the required solution.

Several hill-descending techniques for performing a minimization process are being studied and are yielding dependable results. The question of the complete generality of these methods, however, is still an open one.

E. A LOW-COST PLOTTING BOARD

A hard-copy graphical-display device that can be operated from low-speed, teletype data sources such as those found in time-shared computer installations, is being investigated. A novel feature of this device, shown in Figure 7, is the conversion of incremental binary data into graphical form. This conversion is primarily accomplished by the use of tuned dynamic networks at each axis. Inclusion of mechanical networks, use of certain specific torquers for actuators, and over-all system organization were motivated by a desire to keep cost at an absolute minimum. This objective has not adversely influenced performance or speed of operation. The latter is fundamentally determined by low teletype information rates (100 bits per second) and by upper bounds on the torque-to-inertia ratios of reasonably-sized actuators. Theoretical extensions of the foregoing basic principle have shown that it will be possible to implement curvature control, data smoothing, high-bandwidth plotting and other desirable properties.

F. SIMULATION OF NONLINEAR CIRCUITS

A different attack on computer simulation of electrical networks, from that described in Section A., previously, is being taken in this investigation. Primary emphasis is on nonlinear-network simulation, and network response is obtained through successive solutions and modifications of a matrix which is a mathematical description of the network being simulated. A computer program is now being written in the AED-0 language to establish the network equations and solve them. Later work will include programming of the ESL Display Console for graphically portraying network diagrams and responses to various input signals. The system will be able to handle time-dependent sources and RLC elements whose characteristics are piecewise linear and can be either monotonic or nonmonotonic.

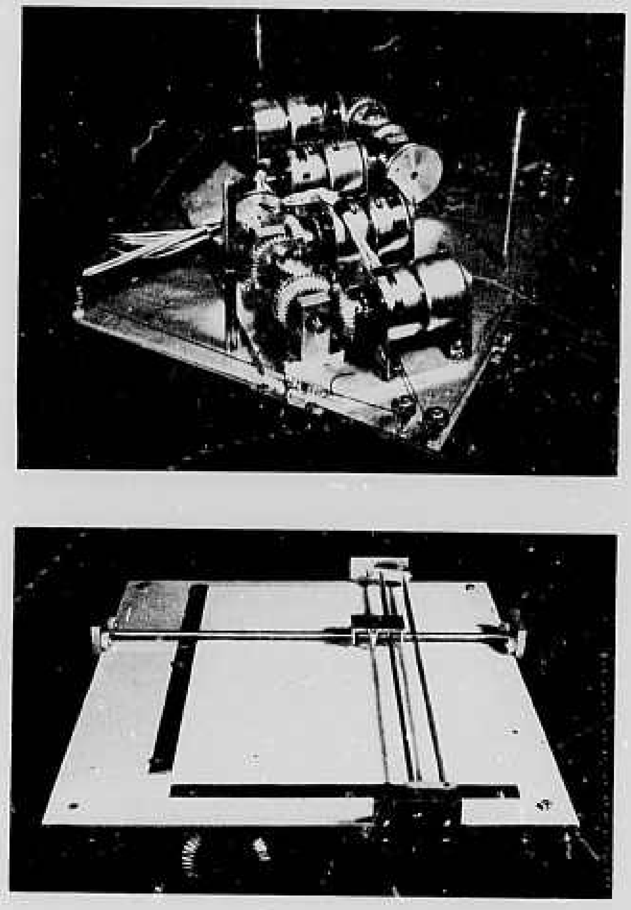


Figure 7. A Low-Cost Teletype-Operated Plotting Board

Accelerometer System Studies - Alfred K. Susskind and Donald R. Haring

A novel approach to obtaining digital data from force-rebalance accelerometers is under investigation for the U.S. Air Force under Contract AF-33(657)-11311. Because the proposed system contains both sampling and quantization, it cannot be treated adequately by analytical techniques. Therefore, experimentation is required, which can be economically done only by means of simulation. The Project MAC Time-Sharing System has been found particularly well suited for this purpose, because of terminal equipment such as the ESL Display Console.

By interplay between theory and simulation, good progress has been made in understanding and evaluating the proposed scheme. The observed system performance is encouraging and the scheme will therefore be investigated in greater depth, gradually removing more and more of the idealizations now being made and also observing the effect of noise. Eventually, a rather complex dynamic system will be simulated, and it is expected that sufficient information will be obtained to act as sound guide lines in the design of an actual accelerometer system.

<u>Simulation of Strapped-Down Naviation Systems</u> - F. B. Hills and J. A. Parisot

In a research program for the Air Force under Contract AF-33(657)-11311, we are studying the computations peculiar to "strapped down" navigation systems, in which inertial sensors (gyros and accelerometers) are rigidly mounted to the airframe rather than being mounted on a stabilized platform. The accelerometer outputs must therefore be transformed from body axes to the coordinates in which the navigation computations are made. We are using the direction cosine transformation. Because of vehicle rotations, the direction cosines are not fixed and hence must also be computed. This is done by numerically solving nine simultaneous differential equations in which gyro data are the independent variables.

Analytical studies made by the group have shown that the major sources of error, insofar as the computations are concerned, are due to the uncertainties that arise in vehicle position and attitude because the input data must be sampled and quantized. The uncertainties arise primarily because acceleration and rotation are not communtative and the components of rotation are not commutative. Hence, not only is the intersample behavior of the input data important, but even more important is the relative intersample behavior between the various input variables. Some analytical studies into the size of the uncertainty have been made; however, the investigation becomes intractable for all but simple flight paths. Therefore, analysis must be augmented by simulation.

Project MAC's PDP-6 was chosen for these simulation studies because of its ability to perform double-precision arithmetic at high speed (46-bit accuracy is required), and its suitability for on-line operational control (e.g., possibility of connecting external equipment, such as a stick to control the flight of a simulated vehicle). Also, macro definitions allowed by the PDP-6 assembly program make it possible to set up desired tests with a minimum of debugging.

A diagram of the simulation is shown in Figure 8. In the box labeled vehicle simulation, the input data, acceleration ($\overline{a}$) and rotation rate ($\overline{\omega}$), are generated. At present, these quantities are generated in accordance with the

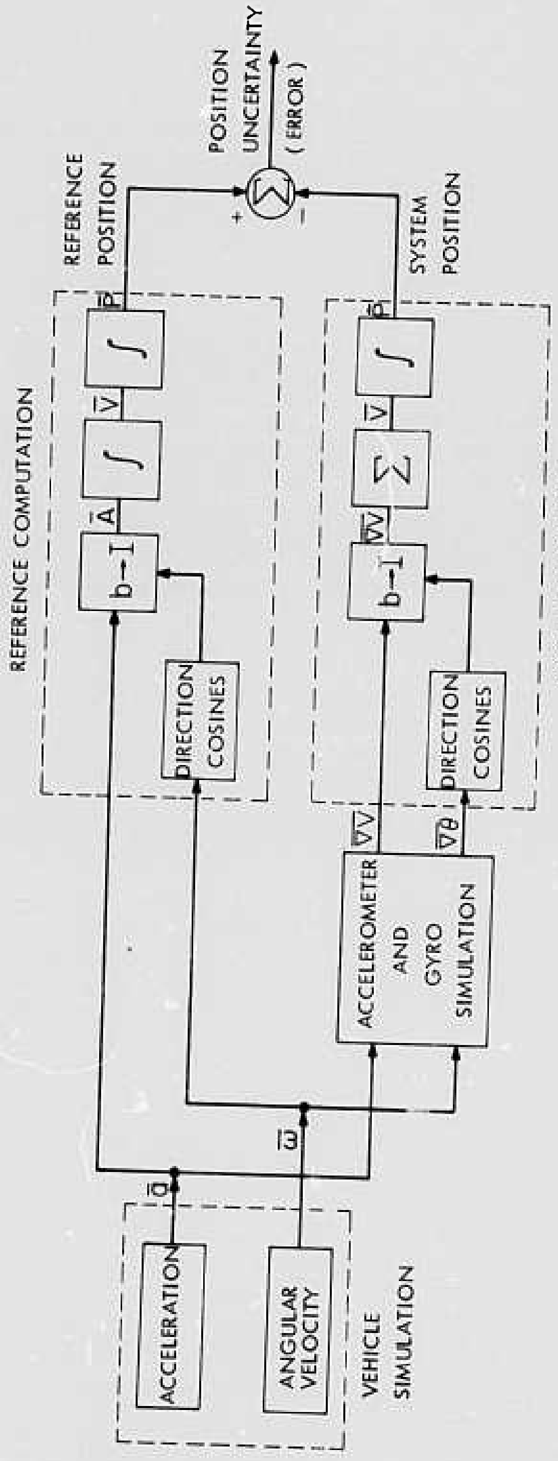


Figure 8. Simulation of Strapped-Down Inertial Navigation System

geometry of simple flight paths. In the future, a simple dynamic model of the vehicle with manual control may be used to allow a search for maximum uncertainty.

Two computations of the coordinate conversion computation must be made: one for the system under test, the other for a reference computation which is considered to be the true solution. The reference computation, like that of the system computation, requires sampling and quantizing of the input data. Analysis has shown that all numerical formulas compute those solutions to the direction cosines that would result if the components of $\overline{a}$ were proportional between samples. Therefore, to make the reference computation more accurate than system computation, the samples for the reference must be taken more often than those taken for the system. This produces a reference solution based on information in the input data not available to the system computation, and hence a meaningful uncertainty can be computed. The high sample rate of the reference computation requires high-speed computation to obtain a close to real-time simulation and long word lengths to limit the accumulation of round-off errors to acceptable values. These two aspects are the major sources of difficulty in programming the simulation.

The block entitled accelerometer and gyro simulation performs the sampling and quantizing done by the inertial sensors. Observe that the outputs are not acceleration and rotation rate, but incremental changes in velocity and angle, respectively.

A set of macros have been written, so that tests can be set up and parameters changed with a minimum of effort. Where several tests are to be run on the same input data (e.g., changes in word length), provision has been made to store pertinent results of the reference computation. Hence, the time-consuming reference computation need be performed only once. Also, an output routine has been written which produces pages of columnar data, each column with a descriptive title. Numbers can be printed in decimal or octal radices, and in fixed-point or floating-point forms.

BLANK PAGE

LIBRARY RESEARCH

Technical Information Project (TIP)

Process Control: Serials and Journals

Search Procedures

Measures of Relatedness

Statistics of Words in Titles

Educational Use of TIP

Use of TIP to Update a Data Compilation

Non-Academic Research Staff

M. M. Kessler
J. Casey
E. J. Dole
M. C. Henneman
I. Johnson
W. D. Mathews
P. M. Sheehan

Research Assistants and other Students

E. L. Ivie
A. N. Kramer

Technical Information Project (TIP) - Myer M. Kessler, William D. Mathews

The M.I.T. Libraries' Technical Information Project is attempting to utilize the time-sharing facilities of Project MAC for literature search and retrieval. The system is a prototype, based on twenty-five physics journals chosen from a reference matrix started by the Physical Review. This literature extends back five years, the average.

For each journal article, we record: location of the article (journal, volume, page), title, authors, institutional affiliation of the authors, citations (journal, volume, page), location of the article in Physics Abstracts, and subject-index information if available from a published source.

This information is edited, compressed, and put on the MAC system as data files. The user may perform search and retrieval operations by means of a special language that consists of several dozen words to control search programs. The simplest form of engagement may be initiated by specifying three commands:

SEARCH	-	defines the range of literature to be searched,
FIND	-	states the items to be found,
OUTPUT	-	defines the nature and content of the computer output.

The SEARCH command may take several forms:

SEARCH ALL	-	will search the entire literature in store,
SEARCH ALL NEW	-	will search the last volume of each journal,
SEARCH PHYREV ALL	-	will search everything in store of a given journal (in this case the Physical Review),
SEARCH PHYREV V. 120 to V. 135	-	only volumes 120 through 135,
SEARCH PHYREV V. 135	-	only volume 135.

The program will search as specified above and detect any item described under the FIND command, which itself has a variety of possibilities.

FIND AUTHOR SMITH - will find all papers that include Smith among its authors.

FIND TITLE CRYOGENICS - will find all papers that contain the word "cryogenics" in the title.

One may similarly define a location,

FIND LOCATION IOWA STATE -

or a citation,

FIND CITATION 1 131 1165 - to find all papers that cite the article in Physical Review (code = 1), volume 131, page 1165. The find instruction includes the logical manipulations of "and/or/but not."

Several other sophisticated FIND commands are available, the most interesting of which is:

FIND SHARE B journal, volume, page.

In this command we are looking for papers in the search range that share a citation item with a given paper. This mode of literature search, called bibliographic coupling, is a powerful tool for information retrieval. The final command, OUTPUT, selects one or more of the options available as output.

The literature and program facilities of TIP are available to all MAC system users, and a monitor program has been written to collect statistics on the type and extent of MAC-TIP utilization. The most extensive use of MAC-TIP to date has been in connection with a book on Plasma Physics, written by Prof. Sanborn Brown, as noted later in this section.

Process Control: Serials and Journals - Patricia M. Sheehan, and Marianne C. Henneman

As libraries grow in size, the problems associated with maintaining controls grow in complexity. Difficulties are most acute in university libraries with scientific commitments, where periodicals represent a large proportion of the holdings.

During 1965, TIP has made a systems study of the traditional manual and punch-card methods currently used to control serials and journals in the M.I.T. Libraries. A feasibility report has proposed using serials and journals as a microcosm for a computer approach capable of:

1) extending the features of the MACTIP information retrieval system to M.I.T. and other regional users,
2) offering alternates to supplement the present card catalogues,
3) providing a two-way communication with the computer, interjecting reports and triggering actions, and
4) performing the usual data-processing functions for administration, accounting, and statistics.

The proposed system will be based on a permanent disk file and a series of programs combining remote console and direct input means for updating and retrieving data. Initial conversion of punch-card data, and a limited retrieval program have been completed. Additional programs are now being written to purify and expand files to meet defined objectives.

As one of the largest files in the time-shared system, the library files need careful program logic to minimize the effects of continually expanding and revising records and items of variable length. A linking technique permits simultaneous use of sorted and unsorted data. Additionally, a multi-level sorter tries to reconcile file data having key breaks, after a comparatively small number of computer characters, with data having a break delayed until the end of a lengthy field. Finally, sorting and retrieval programs expand a variable number of variable-length fields within a record and align each field with the corresponding data in other records. The pragmatic solution at this point is to have the programs determine tentative maximums, and also monitor and adjust the figures as necessary. The extent to which users permit current sources of information to be replaced by computer-produced catalogues and reports will be governed, in part, by aesthetics. For this reason, we hope to ultimately have an off-line printing arrangement, perhaps using Photon or similar equipment, that will be more versatile than standard computer printers.

Search Procedures - Evan L. Ivie

The Technical Information Project (TIP) is based on a measure of the relatedness between documents that is a correlation coefficient based on information theory. This measure was described in the Project MAC Progress Report for 1964 (AD-465-088). A quantitative value is assigned to the "closeness" of every pair of documents in a library, based on the way the library is used.

A program set has been devised which uses this measure to convert a request, consisting of a set of interesting papers and a set of uninteresting papers, into an answer set of related papers. This procedure now allows a user to be very specific, and receive only a few documents in an answer set, by indicating no interest in marginally-related documents; or he may be very general, and obtain a correspondingly larger answer set, by indicating interest in almost everything that is presented to him for a decision. To ensure that the procedure will always converge to a set of related documents, it is necessary that the total internal correlation of the set being formed always increase.

A flexible interaction language has been developed for this system which is a restricted subset of English. Some examples of this language are:

> "Which papers are related to papers by J. R. Jones but not papers using acoustic."
>
> "Print for decision the titles of articles related to articles citing Phys. Rev., Vol. 136, page 22."

Acceptable sentences in the language can be interpreted (or generated) by a 12-state automaton with 8-input word types. The file structure used is "inverted" in nature. The parameters (words, citations, authors, etc.) reside in ordered lists having connectives to other parameters.

Measures of Relatedness - John P. Casey

The volume of journal literature is increasing rapidly. An understanding of the structure of this literature may provide some interesting information about an expanding communication network, and may help in the design of systems to manipulate and order these articles. It is not clear what type of structure will fit journal literature, so tentative beginnings have been made in three different directions.

First, a collection of articles may be considered as a graph, with the articles as points on the graph. An arc connects two articles if they both cite the same article. Questions can now be asked how tightly a subgraph is connected together by its arcs, or how closely connected an article is to a subgraph. These questions are closely related to the difficulties of designing a mechanical system to find a group of articles all related to a given reference.

Secondly, this same network may be considered as a group of nodes connected by bars, with the subject discussed in a given article diffusing like heat along the bars. Coefficients of permeability may be assigned to the bars, and rather complex formulas derived for the amount of "heat" exchanged between an article and others scattered through various journals, to determine which are most closely related to it.

Thirdly, an analysis using the methods of Rashevsky's mathematical biology has been started. From a number of simplifying assumptions about the distribution of useful ideas in articles and related factors, a simple model may be constructed which predicts the likelihood of an article being cited and the time span until an article is no longer mentioned in the literature. This may then be checked against statistical data available through the TIP system. It is hoped that a model which makes reasonably accurate predictions will clarify the general characteristics desirable in a literature-searching system.

We are investigating, for a short distance at least, each of these approaches. Thus far, there have been no results that would show the feasibility, or lack of such, for any of these.

Statistics of Words in Titles - Elizabeth A. Dole

Present studies attempting to describe several characteristics of TIP's vocabulary are based on word-count lists compiled from six years of Physical Review, Volumes 113-132 (1958-1963), and Volumes 133-136 (1964) Series A.

Several criteria are used to assessed total vocabulary, gross word count, and average words per title. The basis of each criterion is an exact definition by formula or by known arbitrary rule. The first criteria developed are those describing what is a "word," and the groups defined are successively rejected from the original unrefined vocabulary. The following steps are planned:

1. A "word" is any string of characters between two spaces. Sample Results:

number of individual words	=	4,826.
individual words times number of uses	=	56,413.
number of articles in sample	=	7,583.
average number of words per title	=	7.43

2. Refinement A. A word is any string of characters between two spaces that does not start with:
   - A. a number
   - B. + or - or =
   - C. any punctuation mark

3. Refinement B. (in progress) A word is any string of characters between two spaces not ruled out by step number two, above, that is not one of an arbitrarily-specified list of words selected for their inherent lack of information context, such as connectives.

   In this step, the list of words is selected and tabulated. The impact of removing these items from the word list GROSS (A) will be calculated to give a new GROSS (B).

4. Refinement C. A word is any string of characters between two spaces not ruled out in steps two and three, above, that bears no basic similarity to any other word.

In this case, words with notably similar bases or roots are grouped, and these counts are compounded in the total figures. The result should be a reduction in the number of individual words, but not in the gross number used for this step.

5. Refinement D. Remove any remaining combination of letters and numbers that is not:

a) an obvious symbol for a chemical compound,
b) an obvious isotope symbol.

6. Refinement E. Remove all chemical compound symbols.

7. Refinement F. Remove all isotope symbols.

8. Refinement G. Remove all unclumped words occurring five times or less. Evaluate the remaining vocabulary. List all words so removed.

9. Comparison I. Compare a portion (or more) of the results above with those gained from an exactly similar list and count from TIP's present full file of 25 journals.

10. Comparison II. Compare search products from equal periods of TIP's library with the library of physics references available through equivalent editions of Physics Abstracts, and through the Physics Abstracts Subject Index Headings List effective at the same time.

In summary, these vocabulary and retrieval studies should show the breakdown, from an information standpoint, of the vocabulary of title words of physics articles as represented in the data base. The studies are distinguished by clearly defined stages. These stages are defined explicitly enough that, once accepted, they could easily be expressed in logical programming for computer processing. The studies so outlined will afford needed information to designers of information files, particularly of inverted title word files, in the field of scientific literature.

Educational Use of TIP - Myer M. Kessler, Sanborn C. Brown, and Irma Y. Johnson

A freshman seminar based on TIP is being offered in the fall of 1965. This seminar is conducted with the cooperation of Prof. Sanborn C. Brown, of the Physics Department, and Mrs. Irma Johnson, Reference Librarian at the M.I.T. Libraries.

Students will investigate the literature of plasma physics using TIP and compare it with the results of a search based on traditional library methods. The course is experimental and is being observed from two points of view:

1. Can beginning students more profitably study a field of physics from current literature references on a computer, instead of traditional text books?

2. Can beginning students compile useful bibliographies and compilations of physics literature from a computer-based system?

Use of TIP to Update a Data Compilation - Sanborn C. Brown

I have been investigating the possibility of writing a report which will demonstrate using a computer to update a compilation of scientific data. Six years ago I published a reference book,* and the present revision is based on bibliographic material contained in the physics literature programmed in the M. I. T. Library's Technical Information Project (TIP). This program is available to all Project MAC users.

Using this system, the attempt is to create material in an open-ended form, so that anyone with access to the computer program can search the literature for material which will appear after the text of my book is printed. As an aid for doing this, references to individual displays of data are given in computer-language form, rather than the usual bibliographic form.

To illustrate the method of keeping up-to-date, let me give an example. Suppose that the reader is interested in the elastic collision cross section between electronics and drypton atoms. He finds a collection of data from L. S. Frost and A. V. Phelps (1964) PHYREV V00136 1538. To find out what further work has been published, assume that new work would cite this paper in its bibliography. The computer program is, therefore, searched to see if this has occurred.

First, the reader must log in to the computer in a manner appropriate to his connection with the computer. Once logged in, he calls for the entire library file with the command TIP and then asks the computer to search for the citation and deliver the information with the command to search ALL and find the appropriate citation, by typing as output the author, title, and identification. The command actually typed into the console is:

```
S ALL from 1964
F CITE PHYREV V00136 1538
O PATI
```

* Basic Data of Plasma Physics, John Wiley & Sons, Inc., New York, 1959.

On the assumption that some new reference was found, the reader then may want to browse further in the associated literature. This is most efficiently done by having the computer search for all papers which share at least one bibliographic reference with the new paper just found. This can, of course, be done against the entire library file, but since the material of "Basic Data" was compiled using this technique of literature search, computer time can be saved by excluding from the search all those papers which already appear in the text. This can be accomplished by a subroutine called "BROWN" created in the process of putting this present volume together. The command on the console to get a shared bibliography based on the program for this volume would be:

```
S ALL BUT NOT BROWN
F SHARE B PHYREV V00160 1234
O PATI
```

Dr. Myer M. Kessler, Associate Director of Libraries at M.I.T. and Director of Project TIP, studied statistically where most of the physics literature was published and developed a list of 25 journals which contained 70 percent of physics papers (Physics Today, 18, 28, 1965). In developing the "BROWN" program for this report, certain journals were further excluded as not being appropriate to experimental gaseous electronics and plasma physics. The list of journals in the "BROWN" program and their program code designation is given in the following list:

Applied Physics Letters	APPLET
Canadian Journal of Physics	PHYCAN
Helvetica Physica Acta	PHYHEL
Indian Journal of Physics	INDJPH
Japanese Journal of Applied Physics	PHAPJA
Journal of Applied Physics	PHYAPP
Journal of Chemical Physics	JCHEPH
Journal of the Physical Society of Japan	PHYSOJ
Physica	HYSICA
Physics Letters	PHYLET
The Physics of Fluids	PHYFLU
Physical Review A	PHYREV
Physical Review Letters	PHYRET
Proceedings of the Physical Society (London)	PHYPRO
Soviet Physics - JETP	SPJETP
Soviet Physics - Technical Physics	SPTPHY

In addition the total TIP program will also search:

Annals of Physics	ANNPHY
Il Nuovo Cimento	NUOCIM
Nuclear Physics	NUCPHY
Physical Review, Series B	PHYREB
Progress of Theoretical Physics (Kyoto)	PROPHJ
Soviet Physics - Solid State	SPSOLS

Another source of data which are not always to be found in the periodical literature are papers published in the proceedings of conferences. A number of these were of importance to the material of this report and these have been especially incorporated into the TIP program. They are Proceedings of the Fifth International Conference on Ionization Phenomena in Gases held in Munich in 1961 (MUNICH), Comptes Rendus de la VI^{e} Conférence Internationale sur les Phénomenes d'Ionisation dans les Gaz held in Paris in 1963 (OPARIS), and Proceedings of the Seventh International Conference on Ionization Phenomena in Gases held in Belgrade in 1965 (BELGRA).

If the reader does not wish to exclude the already search-shared bibliographies in the "BROWN" program, both lists of journals will be searched when "BUT NOT BROWN" is omitted from the console command.

This study is aimed at serving two purposes: one is to bring the "Basic Data" book up-to-date; the other is to provide a computer method of finding new data which appear after the report is issued. The second purpose is accomplished through the citation process, and, therefore, will be effective only if there is some article to CITE. Thus, if no published work has been found which supercedes that published in the original "Basic Data" book, the previous volume's citation will be repeated with its proper reference coding, so that subsequent material may be found by the computer-search method.

This is possible, because, although the TIP program does not contain literature published prior to 1959, the bibliographic coding is complete. If the reader so desires, he may call for the computer to present him with any complete bibliography he desires by calling for "B" as an output command.

In searching the literature, it is sometimes desirable to determine what the common bibliographic references are between two articles found by the "SHARE B" technique. This is provided for in the TIP program by asking the computer to print the linkage between any two papers.

The program has been written so that it is not necessary for a cited reference to be for articles actually available in the program by title, author, etc., but only that it be in the references. Thus, the classic paper on elastic collision was published by Brode in the Reviews of Modern Physics in 1932, long before the time interval covered by TIP, but use of the LINKAGE request will uncover the proper volume and page location of Brode's paper if it has been used as a reference in a recent article.

LINCOLN LABORATORY

Baseband Design of a Unified Carrier

On-Line Data Storage and Retrieval

Compilation For a Digital Differential Analyzer

On-Line Experimentation

Non-Academic Research Staff

J. F. Nolan
A. W. Armenti
F. Belvin
J. W. Bex
M. C. Crocker, III
H. K. Knudson
H. C. Peterson
D. B. Yntema

Baseband Design of a Unified Carrier - Homer C. Peterson

Work on a design-optimization program was started using FORTRAN and batch-processing compiling. The very nature of the equation and the technique of handling were not clearly defined and the infinite limits on the summation made it hard to determine what intervals to investigate. The first runs were so long that they were dumped from the Lincoln computer without progressing far enough to even show a trend to the results. With usually 20 hours elapsing between submitting a run and seeing the results, it became apparent that the problem would either have to be over-simplified or dropped.

At this time CTSS became available to use. The first big step was to recode the program in the MAD language. The program was then modified to be handled a section at a time; where hand comparisons could be made, a rough plot obtained from the teletype, and programming errors readily corrected. Within a week the first outputs were available, while the original had dragged out for nearly a year. Considerable rewrite and modification was necessary, as it became clear that certain sections of the program did not agree either with the model or with the actual world. A new Bessel function subroutine was written, to obtain the desired accuracy, only to find that it worked about half the time. The CTSS loader didn't always properly load a double-precision hardware program, so the program had to be rewritten to be self-shifting if it was started in an odd register.

For nearly a calendar year (but probably less than two man-months of work) the program was in constant change with as many as ten recompilations occurring in one day. During this time, the program was lost only three times due to computer failure. The difference between using a CTSS system and batch-processing was a solution with about two man-months of work as opposed to abandoning the project.

On-Line Data Storage and Retrieval - John F. Nolan

During the past year, the on-line data storage and retrieval system (as described in last year's progress report) was coded, checked, and operated on a variety of sample files. Although limited in data capacity by the disk space allocated to the research problem, the experimental system demonstrated successful implementation of advanced techniques for on-line access and manipulation of formatted data files. The most significant features of the design are:

1. Storage of coded information with the data files describing their organization and content.

2. Representation of this descriptive information by the identical conventions used to store the data files themselves; thus all system commands for searching, cross-associating, and manipulating data files can be used to operate upon the file descriptions as well.

3. The internal use of list-structures to allow unconstrained cross-association of filed data (including, as above, the file descriptions) according to the user's needs.

4. Simple one-word commands and dynamic instructions which allow immediate use without any indoctrination or programming knowledge. (See Nolan, Appendix C)

Compilation for a Digital Differential Analyzer - Harold K. Knudsen

Lincoln Laboratory is constructing a Digital Differential Analyzer (DDA) as part of a hybrid computer facility. A compiler is being written on the MAC system to make it possible for the nonspecialist to generate solutions to ordinary differential equations on the DDA. The inputs to the compiler will be:

1. The differential equation written in a FORTRAN-like language,

2. The initial conditions of the variables, and

3. Estimated maximum absolute values of the variables.

The output of the compiler will be:

1. An interconnection matrix which specifies the internal connections of the integrators and servos in the DDA, and

2. The scaled initial conditions of the problem variables.

The compiler outputs will be loaded directly into the DDA from the MAC computer through a channel consisting of a Model 35 teletype and the LINC computer.

<u>On-Line Experimentation</u> - Joseph W. Bex

During the past year, CTSS was used for the following activities:

1. Searching for sequences with desirable autocorrelation properties, for use in coding patterns for the Arecibo radar antenna in Puerto Rico,

2. Testing the stability of a new circuit design to be used to control the carbon arc for Lincoln Laboratory's new environmental facility,

3. Simulating designs for magnetic orientation systems used on Lincoln's experimental communications satellite, LES.

Exploratory on-line computation, done <u>with the responsible scientist or engineer present</u>, was performed for:

1. Designing antenna arrays for use with the LES satellite, involving the adjustment and testing of various parameter values,

2. Determining the stability or instability of the theoretical model for a new solid-state device which would have unusual high-gain properties.

BLANK PAGE

NON-MIT USERS

Model Testing

Discovery and Learning Techniques and
Programmed Instruction

Generalized Desk Calculator

D. Drukey	Systems Development Corporation
M. M. Flood	University of Michigan
A. Leon	University of Michigan
J. McCarthy	Stanford University
A. Oettinger	Harvard University
S. M. Pizer	Harvard University
A. Ruyle	Harvard University
J. E. K. Smith	University of Michigan
P. J. Stone	Harvard University
L. Uhr	University of Michigan
K. Winiecki	Harvard University

In addition, the Project MAC computer system began to be used by staff of the Bell Telephone Laboratories and of the General Electric Company participating in the development of the MULTICS system.

Model Testing - Merrill M. Flood (University of Michigan)

The purpose of the model testing project is to develop programs and on-line techniques adequate for parameter estimation and goodness-of-fit testing on several stochastic process models. These models derive from the data of various learning and decision experiments with human subjects and animals. We have adapted programs used previously on IBM 7094 computers at the University of Michigan, at the University of California, and at the System Development Corporation, for CTSS use. The programs are now suitable for one specific stochastic model and have been used successfully on a few small test runs.

One set of FORTRAN Subroutines, called LOOK, are part of the general program, ST01. They are an adaptation of a direct search code developed at the Westinghouse Electric Company for the iterative solution of maximization problems with constraints. The LOOK sub-routines will be useful as part of other computer codes and include optimization calculations. Eventually, we hope to make LOOK and other similar optimization codes available to all interested CTSS users. One FAP Subroutine, called PRODUC, within ST01 was written to calculate rapidly and accurately a lengthy product of probabilities. Some of our current programming is devoted to the development of improved codes for this calculation and CTSS is useful in testing such alternative codes for speed and accuracy. We are using these optimization codes to investigate, with experimental data, a few stochastic process models of rat behavior.

Discovery and Learning Techniques and Programmed Instructions - Leonard Uhr (University of Michigan)

We are working on a program that learns how to handle one or two-dimensional inputs, such as language strings and sensory patterns. With the exception of those parts that build up class structures, sections of the program that use previously-learned information to generate the appropriate response to an input have been largely completed and debugged. The sections that learn have been coded, and are now being debugged. We are also continuing tests of the earlier program that handles one-dimensional strings, and we are looking into methods for increasing its abilities to handle different types of problems.

In addition, we have coded a program that turns written materials, such as textbooks and specially-prepared instructional materials, into rough drafts of teaching-machine programs, conveniently edits such programs on-line, and presents the finished instructional program to students. In this program, content material is interspersed with short fill-in questions that lead to forward or backward branches as a function of the student's success or failure in giving a correct or synonymous answer. We can now generate teaching-machine programs in three ways: by compiling from materials prepared according to a simple standard format; by generating, during interactions with the student, from a standard description of a problem domain; and by converting ordinary books into programs. We should now like to combine these methods into a single program that would offer all these options.

<u>Generalized Desk Calculator</u> - Anthony G. Oettinger and
Adrian Ruyle (Harvard University)

Project TACT (Technological Aids to Creative Thought) has been developing the on-line system of Glen Culler, which functions roughly as a generalized desk calculator. With this system, not only can arithmetic operations be carried out on single numbers, as they would on a typical desk calculator, but operations of more generality — such as exponentiation and forward differencing — can be carried out on functions considered as such.

Execution is as quick and easy to initiate as using a desk calculator. Operations on complex functions are also possible, as are plots of functions on a cathode-ray-tube screen, and display of text. There is also a facility for on-line storing of "programs" of operator sequences, which can be executed en bloc as single calculator operations.

The calculator is now developed to the stage where it is a useful tool, for about a half-dozen users who will be variously:

1. Preparing demonstrations, such as prototypes of course lectures in mathematics, and giving calculator exhibitions,

2. Conducting research, such as kidney physiology and frog demography,

3. Exploring the imaginative possibilities of the calculator itself.

BLANK PAGE

OPERATIONS RESEARCH CENTER

Aspects of Integer Linear Programming

Academic Staff

R. A. Howard
P. M. Morse

Non-Academic Research Staff

G. R. Murray

Research Assistants

R. H. González-Zubieta
R. L. Miller

Aspects of Integer Linear Programming - Rómulo H. González-Zubieta

Under U. S. Army Research Office (Durham) Contract No. DA-31-124-ARO-D-209, a primal feasible (all-integer) integer linear-programming algorithm has been developed and programmed, together with a related procedure for obtaining a first feasible solution. Once a feasible solution is found, the algorithm maintains feasibility at each stage, in contrast to other algorithms that have been programmed and are currently available. These other algorithms do not achieve feasibility until the optimal solution is reached. The primal feasible algorithm is based on a particular way of applying the cutting planes previously developed by R. E. Gomory, and on a specific interpretation of their role.

The finiteness of convergence has been established for two-dimensional problems but not for the general case; however, there appears to be at least computational convergence in a considerable fraction of the cases.

In addition, a Generalized Euclidean Algorithm for finding the greatest common divisor for more than two numbers has been defined. The solution of systems of linear diophantine equations is presented in terms of integer linear programming.

BLANK PAGE

RESEARCH LABORATORY OF ELECTRONICS

Introduction

Dynamics of Active Plasma Systems

Dispersion Relation for Hot Plasmas

Plasma Dispersion Relations with Infinite Roots

Stability Analysis of Dispersion Relations

Use of CTSS in a Plasma Physics Experiment

Analysis of Speech

Simulation of the Human Larynx

Articulatory Events of Speech Generation

Grapheme-to-Phoneme Translation of English

Programming Support and Development

RLE Statistics on CTSS

Sorting of Personnel Records

COMIT

Academic Staff

G. D. Bernard
A. Bers
T. H. Crystal
W. D. Getty
M. Halle
H. A. Haus
J. M. Heinz
L. D. Smullin
D. E. Troxel
V. H. Yngve

Non-Academic Research Staff

C. M. Bosche
A. Congleton
J. L. Darlington
R. S. Fabry
G. Fratar
M. S. Kannel
E. R. Lewis
V. E. McLoud
M. M. Pennell
R. L. Rappaport
E. River
R. A. Sayers
K. A. Szabo

Research Assistants and other Students

R. Bartsch
W. Y. Chang
J. A. Davis
T. J. Fessenden
R. V. Harris
T. R. Hofmann
H. Ishii
M. E. Kaliski
B. R. Kusse
F. F. Lee
M. A. Lieberman
M. M. Lind
D. T. Llewellyn-Jones
J. D. Mills
E. A. Robertson
H. M. Schneider
C. L. Seitz
C. E. Speck
S. Spitzer
B. T'sou
R. N. Wallace

Introduction

This interdepartmental laboratory provides facilities for academic research in three categories, designated as general physics, plasma dynamics and communication sciences.

Major support for the research is provided by the Joint Services Electronics Program of the Army, Navy and Air Force as well as the Atomic Energy Commission, the National Science Foundation, the National Institutes of Health and the National Aeronautics and Space Administration.

As indicated by the following section, a substantial number of RLE research projects have received invaluable assistance from Project MAC facilities. These faculty and student research activities span a wide range of scientific and engineering subjects.

Dynamics of Active Plasma Systems - Abraham Bers

During the past year we have made extensive use of the MAC computer and the ESL display system to solve a large number of difficult and important problems in plasma stability. The complexity of these problems is such that without direct man-computer interaction many of these problems would not have been attacked. The basic problem of plasma stability to small-signal perturbations was described in our progress report last year. The most significant programs developed during the last year for our problems are summarized in three reports that follow (M. A. Lieberman; C. Speck and M. Pennell; and J. D. Mills and A. Bers).

Our project has brought a large number of our students in contact with the Project MAC facilities. The thesis research of these students has ranged from computer-display system programming to computer stability analyses of hot, gaseous plasmas, and plasmas in solids. From our group, a total of about a dozen graduate students, three faculty members, and two post-doctoral visitors have made use of the Project MAC facilities during this past year.

Dispersion Relation for Hot Plasmas - Michael A. Lieberman

We have written a computer program that finds the zeros of a transcendental dispersion function $\underline{D}(\underline{\omega}, k, \dots)$ in the complex $\underline{\omega}$ plane. Given the complex frequency $\underline{\omega}$, the wave number k, and any other parameters that a user may desire, he must provide a subroutine within CTSS that computes the value of $\underline{D}$. This subroutine may be written in MAD, FORTRAN, or FAP.

In operation, the program continuously steps up k by an increment Δk and, at each step, finds a zero, $\underline{\omega}_n$, of the dispersion function $\underline{D}$:

$$\underline{D}(\underline{\omega}_n, k+n\Delta k, \dots) = 0.$$

The zero, $\underline{\omega}_n$, is found through the construction of a grid of values in the complex ω plane around an initial guess ω_{gn} as follows:

$$\underline{\omega}_{go} = \text{provided by the user}$$

$$\underline{\omega}_{g1} = \underline{\omega}_o$$

$$\omega_{gn} = \underline{\omega}_{n-1} + \Delta k(\underline{\omega}_{n-1} - \underline{\omega}_{n-2})\ n \geq 2.$$

The program evaluates the real function $F = |\,D(\underline{\omega}, k+n\Delta k, \dots)\,|^2$ for every point on the grid. Whenever a minimum of F is found at ω_n, the grid is refined several times until ω_n is given to three significant figures; then, $\underline{D}$ is checked to verify whether its real and imaginary parts change sign in the neighborhood of $\underline{\omega}_n$; and, if they do, the zero is printed and k is stepped up. Thus, the zeros of D in the complex $\underline{\omega}$ plane are computed as a function of the wave number k.

Without time sharing, a solution of this problem would be prohibitive; with it, a user may specify and alter at will the grid size and spacing in the complex $\underline{\omega}$ plane, the wave number k and its increment Δk, and all other parameters. If at any step a zero of $\underline{D}$ is not found, the program requests the user to change the grid size, spacing, and location in the complex $\underline{\omega}$ plane. To help the user, the program will print the values of $\underline{D}$ and F at the grid points if they are requested.

Plasma Dispersion Relations with Infinite Roots - Carlton E. Speck and Martha M. Pennell

In the study of a particular class of plasma waves, the following equation must be solved for its roots, VC = RVC + jCVC:

$$1 - (VPC)^2 \left\{ \frac{1}{2} \sum_{n=-\infty}^{\infty} \frac{J_{n-1}^2(p) - J_{n+1}^2(p)}{(VC - n)} - \frac{\beta_\perp^2}{p^2} \sum_{n=-\infty}^{\infty} \frac{n^2 J_n^2(p)}{(VC-n)^2} \right\} = 0$$

We had to determine those values of $\beta_\perp$ and VPC which had ranges of p that resulted in roots with a negative imaginary part. Since the equation has an infinite number of roots and is not expressible as a polynomial, we used Newton's method to locate the roots. However, Newton's method requires an accurate guess of the root location. To obtain these guesses, we wrote subsidiary programs for the approximate portion of the root. These approximations derive from an approximate equation in only those terms of the infinite sum which are large when $|VC| \sim n$. With such "guessing" programs an operator can find the required approximation for the more exact Newton's method and, by varying VPC and $\beta_\perp$, can rapidly search the VC - p plane until he finds the approximate regions of the negative imaginary part of VC.

<u>Stability Analysis of Dispersion Relations</u> - James D. Mills and Abraham Bers

A program has been written, checked out, and is now being used by members of the Beam-Plasma Group of the Research Laboratory of Electronics in the stability analysis of plasma dispersion relations. The program makes use of the ESL Display Console for graphical outputs. The time-sharing system facilitates man-machine communication for changing parameters and equations in the analysis.

The motivation for the program came through the formulation of the Bers-Briggs stability criteria. These criteria enable one to analyze the dispersion relation of a plasma (or other) system, to determine and classify the kinds of stable and unstable wave responses which are inherent to the system. The technique involves the general problem of mapping a function of a complex variable. The dispersion equation of the physical system to be analyzed, $D(\omega, k) = 0$, relates the two complex variables which are $\omega = \omega_r + j\omega_i$ and $k = k_r + jk_i$. One then maps lines of constant ω_r or ω_i into the complex k-plane or lines of constant k_r or k_i into the complex ω-plane.

The program restricts the user to polynomial-type equations of no more than degree twenty. The coefficients of the terms of the equation are specified simply in a subroutine provided by the user. He then translates this subroutine using the MAD command. It is then loaded together with the display program which queries the user concerning various parameters. When these are specified, the program makes all of the calculations necessary to the mapping and displays the results on the Display Console. In Figure 9 for example, the user specifies the lines indicated on the complex ω-plane (a) and he sees the complex k-plane (b) plotted on the display.

The program has been used in various stages of development since April, 1964. A joint RLE - Project MAC technical report is being prepared for publication. (See J. D. Mills thesis entry in Appendix B.)

Figure 9. Mapping of Dispersion Relation $D(\omega, k) = 0$

Use of CTSS in a Plasma Physics Experiment - David T. Llewellyn-Jones

This work combines automatic data acquisition and processing with a laboratory experiment in plasma physics which requires rapid feedback between experimenter and his apparatus. The last part of this work will establish an interface between a digital voltmeter, connected to the laboratory experiment, and a teletype, connected to the MAC computer. Since a teletype has been installed beside the experiment, the experimenter has been able to simulate actual runs and at the same time, to act and manually type the data into the teletype as it is displayed on the voltmeter. In this way, speeds of approximately one, three-digit data-point in every 2.5 seconds are possible. Since the experiment can run at speeds up to about one data-point-per-second, realistic simulation is only possible for slow runs.

In simulation runs, the .TAPE. feature of CTSS is very useful. This feature immediately creates a permanent data record before the data reaches the program, and it has all the useful correction and substitution facilities afforded by the INPUT mode. When the data was read directly into the program and the necessary correction and substitution facilities had to be written into the program itself, the program became cumbersome and slow. After installation of a reliable automatic recording system, the data will be read directly into the program and the program itself will automatically create a .TAPE. of the data as a record.

Analysis of Speech - John M. Heinz and Eleanor C. River

As part of a study on the constraints which the vocal organs impose on a speech signal as a result of their physical configuration, we are trying to model the process of speech production from its description by lateral cineradiographs up to the resulting acoustic signal. Four stages exist in analyzing these lateral cineradiographs and relating them to the corresponding quasi-static acoustic spectra: 1) obtaining from the film a frame-by frame tracing which includes all of the relevant detail, 2) devising a reference coordinate system for quantitative measurements which can be precisely defined in terms of stable landmarks readily observable on the film, 3) devising, from a knowledge of the structural anatomy, a set of transformations by which the acoustically-relevant cross-sectional area may be inferred from the lateral measurements, and 4) calculating an acoustic spectrum from a specification of the cross-sectional area as a function of distance along the vocal tract midline. The programs that have been written to aid in the construction and testing of the models of speech production correspond to parts 3) and 4) above.

A preliminary version of the midsagittal-to-area function transformation program is now in operation. With this program, we can preset the cross-dimension-to-area function transformation for a particular talker, either by reading in data tables prepared from anatomical measurements of him or by specifying certain parameters which determine an analytic description of the transformation. One of the objectives of the present study is the investigation of various possible types of analytic descriptions. Once the program has been preset for the talker, cross-dimension data from the films are read in. The program then calculates both the cross-sectional area as a function of distance along the vocal tract midline and the effective vocal-tract length.

We have written a second program which accepts cross-sectional area functions derived by the first program as input and, with a Webster's equation approximation to an acoustic description of the vocal tract, determines the corresponding acoustic output spectra. We have then compared a computed spectrum with the spectrum of the actual sound produced at the time of the x-ray filming. Such a comparison tests the validity of an investigated model.

Simulation of the Human Larynx - Thomas H. Crystal

During the past six months, we have developed a set of programs for simulation of the behavior of a human larynx during the production of speech sounds and for control and display of the results of the simulation on a remote typewriter console. The programs still need some additional modifications, but have produced some interesting results.

The larynx itself is represented by two systems. A mechanical system, which models the mass and tension of the vocal cords, is used to determine, as a function of time, the opening between the vocal cords and air flow through the vocal cords. The parameters of this system depend on the vocal cord opening as determined by the mechanical system. Difference equations that describe these systems are iterated successively to produce the simulation. In addition, the program contains provisions for the connection of any of a number of vocal tract and sub-glottal models to the larynx model. These systems should together provide a complete picture of larynx activity.

A feature of the simulation programming is the ease with which the progress of simulation may be controlled and its parameters varied. This progress, together with various plotting, printing, and frequency analysis routines, is controlled through simple, one-word commands. Thus, a user may perform a simulation, examine the various waveforms generated, perform Fourier analyses on any particularly interesting waveforms, change the parameters controlling the simulation dependent on the results of the previous simulation, and simulate again. Because of extensive control facilities programmed into the model, full advantage of time-shared computation can be obtained in this simulation problem.

Articulatory Events of Speech Generation - William L. Henke

This project applies computer information-processing techniques to the study of the articulatory events of human speech generation. Our data, which consist of cineradiograph films of human speakers, together with their corresponding phonemic input and acoustic output, are studied and modeled mechanically for the purpose of understanding the human speech process, in particular the transformation between a discrete phonemic input and a continuous acoustic output. As we are dealing with spacially-distributed structures, we require the input/output facilities afforded by the ESL Display Console. Since the information-retrieval and modeling of these data depend upon techniques of on-line interaction of a user with a computer and non-textual types of information, part of our work is the development of such techniques. Three classes of programs have been developed to the present: 1) cineradiographic graphical data input, display, and analysis; 2) implementation and display of a model of tongue action; and 3) a very general solution for the acoustic transfer function of the vocal tract, with associated displays and inputs. They are currently disjointed, but further work will increase the communication between them so that they may be used singly or together, depending upon the needs of the aspect of speech being studied.

One part of our work involves the input, storage, and display of a graphical representation of the mid-sagittal plane of a human vocal tract in action. The MIT Speech Group has recently made frame-by-frame tracings of projections of cineradiographic films of speakers. These tracings are fortunately about the size of the ESL Display Console scope face, so that with tracing paper affixed to the face of the scope, the light pen can see the tracking cross through tracing paper. We enter points and connected line segments for various anatomical structures and landmarks, together with identifying information, into a dynamic data structure. At the end of the input for one frame, the data are edited and written out on the disk. The first generation display program searches the file for desired frames and simultaneously displays up to three at different intensity levels. The organization of this file and the specification and recording of complementary acoustic and phonemic data, which will be suited to analysis by visual comparison, has prompted our studies on graphical input.

We are developing a model of articulatory action which will distinguish between operators that are associated with inputs representing linguistic categories, and states that are associated with the responding articulatory structures. Considering the behavior of the tongue as a responding structure, our work develops general subprograms for the display of graphical and textual data and constructs a time-domain simulation model which stops at any desired time and displays the present state. With this model, a user can then compare the state with x-ray tracings and modify some of the parameters of the model.

For acoustic comparison, an algorithm which calculates the frequency-domain response of the vocal tract is desirable. We have written one for general situations which calculates both the complex spectrum from input data of the area function and the loss as a function of frequency, position, and area. From the complex spectrum, it further calculates poles of the transfer function. Since all this information is displayed on a CRT, the volume velocity distribution at any frequency is available. Programs developed by other members of the Speech Group to obtain area functions from mid-sagittal plane representations of the vocal tract will combine with the programs discussed above.

<u>Grapheme-to-Phoneme Translation of English</u> - Francis F. Lee

It appears that when an English-speaking person "sounds out" a printed English word or an English-like non-word, he makes use of several often mutually-supplementary processes, namely, recognition of the pattern as a whole, identification of certain affixes, in particular the suffixes, identification of compound words (e. g. Whitestone as white-stone, not whitest-one) and a mapping of graphemes to phonemes employing the graphemical context, with the last process occurring at the latest or innermost level.

The idea of a mechanical procedure, translating a printed English word into its phonemic representation using only a single and small exception list, runs into immediate trouble. By using multiple exception lists, consulted when necessary, the total number of exceptions and the list of context sensitive rules can be substantially reduced. The extent of the necessity of affix identification is being studied.

It appears that the presence of mute e in medial position due to word compounding, such as spacecraft can only be detected by some sort of table-lookup procedure.

The 18,000 more often used English words (from Thorndike and Lorge) have been processed to show the grapheme-to-phoneme cluster relationship in graphemic context. This body of data has been used for the creation of exception lists. It will be used for the decomposition of complex clusters and the formation of rules for the previously-mentioned innermost level.

Programming Support and Development - Martha M. Pennell

The Computation Group has provided both programming support and development for users of the time-sharing system. Our program QUIXOT, written to alleviate the programming load of the Computation Group, has been operational for almost an academic year. A user need only type an algebraic expression for the coefficients, and QUIXOT writes, compiles, loads, and runs a program to find roots of a polynomial of degree twenty or less.

Encouraged by the success of this program, the Computation Group modified QUIXOT so that the user can type in his coefficients as one expression, rather than as real and imaginary parts. Because MAD, the source language for this program, does not provide for complex arithmetic, a subroutine call must perform each complex arithmetic operation. Also, we incorporated a compiler-writing program into the QUIXOT program, which saves the user some initial algebra. An outgrowth of this program is COMCAL, a routine to evaluate an expression in complex arithmetic. Similar in operation to QUIXOT, it first queries the user for his symbols, parameter names, and algebraic expression; then, with the command-programming feature of time-sharing, it writes, compiles and executes a program to evaluate the expression.

RLE Statistics on CTSS - Martha M. Pennell

I cite examples which might give some idea of the time difference involved in a programmer's speed with CTSS versus batch processing. They are taken from the work of RLE Computing Group.

A mathematics major with one summer's FORTRAN experience was hired as a scientific programmer. She was given a short demonstration of CTSS — compute A from the formula A = B + 1 where B is input from the teletype — and told to duplicate it where B is an array of N numbers to be given at execution time. She did so in about four hours. The next day this person was given an RLE problem, and the calling sequences for the two necessary subroutines (a Bessel function and roots of a polynomial), and she was told to code it in MADTRN. Within three days her program produced results. At first it contained errors, but she had set up the subroutine linkages correctly and had made no errors in its logic. Her errors lay in the algebra for computing the sum. This initial coding was done with a minimum of supervision. When I later asked how long she thought it would have taken her on batch processing, she said a month, because her errors were due to her unfamiliarity with FORTRAN and a series of programming blunders, such as keypunching errors and omission of control cards or subroutine decks.

Another user has five years of experience in FORTRAN and one year with MADTRN. The majority of programs she has had to write on CTSS can be coded and debugged, on the average, with a two-hour session at the console. These same programs, if run and checked out under batch processing here at M. I. T., would take at least four to six days. These programs involve a main program and possibly two or three subroutines of about 40-50 statements each. These programs have found roots to polynomials, evaluated Bessel functions, performed numerical quadrature, and so on. She finds that with programs of this length, three or four tries usually debugs them. If you take into account machine errors, bad tapes, lost output, this usually means about a week in a batch-processing scheme.

This person also decided to learn MAD. The problem she chose to code was a tree sort. Given the people in her office, their sex, their marital status, their laboratory affiliation, and their telephone number, the problem was to to write a program to sort them, grouped alphabetically, according to one or more of their characteristics. The MAD program was about 40-50 statements long and took six tries before it finally compiled, because the programmer was not completely familiar with the punctuation and rules of syntax. Two more tries debugged the program logic. This work was done in about two hours of console use.

Sorting of Personnel Records - Robert L. Rappaport

The RLE personnel roster includes approximately 700 people, who are classified by several categories: position, research group, rooms occupied, telephone extension, and so on. For example, an associate professor of the Electrical Engineering Department who works in the Plasma Physics Group must be listed under each of these files. At present, these files are brought up to date and sorted manually.

We have written a computer program to automate this organization. The program operates on a masterfile which contains all the essential information for every member of RLE, so that we need make correction only once. Sorting can be done on any of the items entered. A second program prints an alphabetical directory of all RLE personnel. We have checked out both of these programs for a limited number of names and will shortly extend them to the entire RLE roster.

COMIT - Victor H. Yngve

During the year we maintained the COMIT system as a standard programming language available to users of MAC, and we further developed the language so that a new system, COMIT II, will soon be released for use in MAC. COMIT II includes many new features and conveniences for the programmer, while any program that works in COMIT will also work in COMIT II. We also developed a program VEDIT, which is written in COMIT and is especially useful in the writing and editing of COMIT programs. It introduced a number of new features that have found their way into recent edit programs.

Jared Darlington further developed his set of COMIT programs that check the validity of arguments expressed in English. These programs are unique in that they incorporate several levels of analysis of the input argument to determine the lowest level of logical analysis sufficient to prove the argument valid. In addition, a program in COMIT which partially translates programs from FAP to GE635 GEM has been developed by two undergraduate students.

SCHOOL OF ENGINEERING

Computer-Aided Design
On-Line Mathematical Analysis
Derivation of Preliminary Ship Lines
Time-Sharing Reactor Code System
A Stress-Analysis Program
Stress-Analysis Conformal Mapping
Computer-Aided Teaching of Dynamic Systems Behavior
Automatic Network Synthesis
A Computer Model of Kinesthesis
Computer Solutions for Boundary-Value Problems
Algebraic Expression Compiler
An Algorithm to Aid Logic Design

Academic Staff

G. S. Brown
J. W. Brackett
S. A. Coons
K. F. Hansen
R. Kaplow
H. S. Mickley

Non-Academic Research Staff

R. Bayles
G. L. Benton
C. Garman
T. Lerner
R. Leopold
J. W. Lucey
G. A. Maley
T. Pitidis-Poutous
I. C. Pyle

Research Assistants and other Students

W. C. Albertson
A. E. Armstrong
G. Grover
N. B. Heubeck
R. Hodges
B. V. Koen
J. Ku
H. Levin
P. C. Lynch
Q. L. Miller
D. B. Murton
R. Parmelee
G. Piotrowski
R. C. Rosenberg
K. Rosenfeld
R. Sayre
E. H. Sibley
D. Somekh
S. L. Strong
C. C. Tillman, Jr.
M. Weinberger
J. I. Weiner
P. A. Wieselmann

Computer-Aided Design - Charles Garman, Richard Bayles, and Harold Levin

During the past year, Mr. Bayles and Mr. Levin were programming the section of the CTSS supervisor responsible for maintaining the display on the ESL Display Console. Since the design of the software for the display system requires a fairly rigid and highly artificial structuring of display data, operation of the first version of the supervisory program soon showed defects in this design.

We began in August, 1964, to redesign the interface between the user and the display system and succeeded in developing a new supervisory program. Among its features were: the recognition of the slave display, which had been added to the display console, so that two users can maintain different displays simultaneously; the addition of a system which assigns real-time tasks to various analog and digital inputs of the display console and to different sections of the image; and the addition of subroutine pictures, which facilitate display of similar graphical objects. This project was completed in December; and, with the help of Michael Brescia of the ESL Display Group, we have continued to modify and improve the system.

At the end of December Mr. Bayles began programming for the Disk-Drum Strategy module that was to be incorporated into the revised File Control System of the CTSS supervisor. This work was continued and has since been completed by Mr. Garman. He was also involved during the previous summer in programming a system that generated and displayed on the ESL console that were surfaces derived from an algorithm of Professor S. A. Coons. These surfaces are completely general in nature and may model, with remarkable precision, objects as diverse as boat hulls, an octant of a sphere, adjacent planes of a regular tetrahedron, or a section of an airplane fuselage. In December, he began work on modifications and extensions to the CTSS 'ED' command, which is a context editor for BCD card-image files and he is reprogramming the input-output section of 'ED' to conform to the new file-control system.

On-Line Mathematical Analysis - Roy Kaplow, Stephen L. Strong, and John W. Brackett

With our MAP system a user with little knowledge of computers or programming can solve a wide variety of mathematical problems expressed in normal mathematical terminology. MAP contains many interrelated programs, only a small fraction of which may be executed in response to any one request. The language consists of functional equations and an expandable group of commands. All communications can be transmitted through a standard Project MAC console.

The system evaluates one-dimensional functional equations that involve constants, vectors, library functions and ordinary arithmetic operators. The form used is identical to normal notation, except that multiplications only implied in the first form must be specified in the latter and functional operations, such as sine and cosine, must be designated by a terminating f. MAP automatically generates the q(x) vector at equal intervals over any range in x specified by the user. If the range and interval in x, the values of the constant a, or the values of the vectors y (x) are not already specified, they are requested by MAP and may then be typed in under the control of context editor.

Commands in MAP correspond to requests for operations upon functions, such as q(x), or upon groups of functions, which cannot proceed in the point-by-point fashion of equations. With such commands at his disposal, the user need not remember the parameters required for the operation.

As an example, the typing of the single word "integrate" will cause MAP to: ask the user if he desires to integrate between two constants, the full range of an independent variable, or some combination of constant and variable limits; ask the name of the function to be integrated; the name to be given to the constant or function answer; and, if necessary, the values of the limits. As with equations, if the function to be integrated is not already defined, its values will be requested and may be typed in with a context editor. The sophisticated user may prescribe all of the necessary information, regarding options within the operation and parameter names or values, and avoid unnecessary communication. The following operations are now available: integrate, differentiate, convolute, Fourier transform, change of basis, and least-square fitting.

In addition to the equations and simple commands discussed above, other commands are available which facilitate problem solving and which extend the range of MAP. These are briefly described below under five separate classifications.

A. DATA HANDLING
These commands provide for off-line input of blocks of data, for the deletion of data from mass memory, and for the saving of data and programs. All information is handled by its symbolic name, and the user need not be concerned with computer techniques.

B. FUNCTION HANDLING
The "minmax" command changes the range and interval for which an independent variable or function is defined. The "select" command manipulates a portion of an individual function.

C. OUTPUT
The "print" command is used for all output other than system-generated messages and questions. The format of printing may be specified by the request of a user. When he so desires, a user may request that only part of the tabulated range of a function be printed, together with an interval in the independent variable.

D. COMMAND SEQUENCE CREATION AND EXECUTION
The "run", "create", and "edit" requests define a sequence of MAP statements and store them in mass memory for subsequent use. "Create" allows typing in a sequence of commands under control of the context editor. The sequence is assigned a name and executed by "run" and the assigned name. "Edit" edits a sequence of commands to correct errors or to substitute variable names when the sequence is executed by a "run" command. A user who has not learned a complex language, such as FORTRAN or MAD, can thus create a programming structure for his specific application and call it by its symbolic name.

E. EXECUTION OF MAD PROGRAMS

With "Execute," a user can write his own command in the MAD language and use it within the MAP system. The existence of this command makes the system easily expandable and should encourage users to add to the system. After the user has written a MAD program, such as "root," he need only type "execute root" to run his program. As an aid in the creation of his own commands, a large number of system subroutines are available to the facilities of the MAP system. Other subroutines accomplish such tasks as: loading any function, previously defined within the MAP system, from mass memory into any array specified in the user's program; obtaining one or more values of a function from mass memory, and storing them as the values of any variable in the MAD program; creating a MAP function from an array in the user's program; and executing any CTSS command. All of the necessary subroutines are automatically loaded along with the user's program when the "execute" command is given.

During the past year the MAP system was made available to a number of users of the MIT Computation Center. Many of the alterations and additions provided during this year are the result of our experience with those users. Plans for the forthcoming year include extension of the system to functions of more than one independent variable, matrix operations, differential equations, and addition of graphical input and output.

Derivation of Preliminary Ship Lines - Theodore M. Pitidis-Poutous

The object of this investigation, supported by Bureau of Ships Contract No. NObs 90100, was to develop a suitable procedure for the mathematical derivation of preliminary ship lines, for a new design, by reference to the lines of a parent ship. An algorithm has been formulated to determine a new curve or surface, in the form of a polynomial, resembling as closely as possible another graphically-specified curve or surface, and satisfying a prescribed set of constraints. The computations involved are facilitated by the use of special polynomials.

Various ways of applying this algorithm to the delineation of preliminary ship lines have been investigated. In particular, a procedure has been formulated whereby the preliminary lines for a new design can be obtained in two steps. First, the parent form is fitted with a polynomial expression. Then the surface equation for the new design is obtained by modifying the expression for the parent hull to conform to the new prescribed hull form parameters. As an example, the hull form of a trawler was first fitted and then modified to conform to new values of prismatic coefficient, maximum area coefficient and waterline coefficient.

The first step involved development of a suitable mathematical model of the parent form by fitting the hull surface to a single polynomial expression, on the assumption that the surface could be considered as smooth. It is possible to express the hull form in terms of Legendre, Chebyshev, or ordinary polynomials, whichever is advantageous in the particular case, while the fitting technique allows the number of terms of the expression to be chosen according to the required accuracy.

In this connection, Chebyshev polynomials were found particularly suitable. The additional labor involved in drawing a Chebyshev body plan for the preparation of input data was thought to be justifiable, since better fitting could be obtained with a smaller number of terms. The approximations obtained were deemed acceptable for preliminary and feasibility studies. Undesired oscillations, which are unavoidable with high-degree polynomials were minimized with the fitting technique and appear to be of tolerable magnitude. Polynomial approximations to ship hulls could perhaps be improved further only by transforming the hull surface to a more "respectable" one before the fitting is undertaken.

Once a good approximation to the parent hull form was obtained in terms of polynomials, the second step involved deformation or modification of the parent form according to new form parameters.

The use of additional constraints were useful in making radical changes to the parent form. It is possible, at least in principle, to incorporate a bulb by applying suitable constraints to the forward sections, and by suitably modifying the sectional area and designer's waterline curves. In the trawler example, nine such constraints are necessary for each section in the region of the bow, so that the corresponding parent section is ignored. Further investigation concerning the use of the constraints already employed, or possible additional constraints, will be necessary to critically evaluate this procedure.

These remarks apply to the trawler case examined so far, as full ship forms have not been investigated. Preliminary experience with very full sections indicate that a few more terms would be required for approximations comparable to those obtained for the trawler. Further investigation is necessary, however, before definite conclusions can be drawn with respect to full forms. (See Pitidis-Poutous, Appendix C.)

Time-Sharing Reactor Code System - Kent F. Hansen, Ian C. Pyle, Billy V. Koen and Q. L. Miller

The purpose of this work is to study the use of time-sharing in Nuclear Engineering education and research. In a question-and-answer session at the console, a user derives input data for any nuclear code, in such a way that he need have no knowledge of computer programming at all. In the past year, we have completed a general-purpose input/edit program that obtains input data from the console and numerous nuclear codes. The entire collection of programs is called the TREC system (Time-Sharing REactor Codes).

The TREC system operates as follows: when the user requests a particular code, TREC obtains the appropriate files and certain satellite routines that read in the input data. The input/edit program asks the user for each item of input in a language natural to the problem. The user may review, change, or correct his data. After all data is properly given, he may request the code to start; TREC then transfers data and control to the nuclear code. At the conclusion of the calculation or after an interrupt, control is returned to the input/edit program for possible changes in data. At the conclusion of the run, the user may select another code.

The function of the routine TREC is to ask the user what code he wants, determine if the code is available or not, and, if it is available, locate the symbol table and test routine for the code. The symbol table itself is a list of encoded variable names and descriptive phrases that define each variable. The data is stored in a linear array for delivery to the nuclear code itself. The test routines are used to check all input data. Occasionally a block of data may be standard information, such as neutron delayed-yield fractions for fissionable species, in which case the test routine also provides the standard data. Upon completion of the input/edit phase, the input data array is written out in a pseudotape for delivery to the nuclear code. Within the symbol table, the name of the file containing the nuclear code is stored for transfer of control.

Finally, the routine QUIN is loaded. QUIN scans the symbol table and prints comments and questions on the console. Thus, the symbol table contains the names of all input data along with additional details. The test routine is used to test the validity of each piece of input data. If an impossible or inappropriate value is given, the test routine prints a comment to that effect. Such details as decimal-to-binary conversion are also contained in QUIN.

Several nuclear codes have been prepared and are now routinely available. A brief description of each is given below.

1. KINET solves the infinite medium-kinetics equations with a variable number of delayed neutron groups.

2. DIFFUS is a one-dimensional, multigroup-diffusion code. Its program handles up to 16 groups, 5 regions and 5 isotopes per region, and also a maximum of 100 space points.

3. CROSEC is a code that computes group-average, nuclear cross-sections.

4. The Cross Section Library is not a code but rather a large block of nuclear cross sections of many isotopes. The data is encoded in a form used by CROSEC.

5. DNPORT is a double P_N transport code. This code analytically solves the transport equation for one-group slabs. Although it is written and debugged, the symbol table is not yet completed.

A Stress-Analysis Program - Richard P. Parmelee

During the past year, I finished and tested a two-dimensional stress-analysis algorithm for thin domes or shells, and then extended the algorithm to handle three-dimensional stress analysis of sculptured bodies, such as crankshafts. I completed this three-dimensional program in March of 1965 and since then have solved test problems.

The object of my research to find a method for stress analysis of virtually any geometric region, and then implement this method with a computer program that will permit a user who knows nothing about computers to define his problem and obtain accurate answers. Although I found an algorithm suitable for analyzing a large class of problems, I had problems with the implementation of this algorithm. The availability of a time-shared computer made possible the coding and debugging of this program in a reasonable time, as well as the guiding and error checking of the user input.

Although I did some graphical display work with the ESL Display Console, the input, output, and program control was, for the most part, restricted to a teletype.

Stress-Analysis Conformal Mapping - Paul A. Wieselmann

Fundamental to the solution of plane-stress-analysis problems by complex variable methods is the generation of a function to conformally map the unit circle, or upper infinite half-plane, onto a region described by the physical domain of the problem. In general, only approximations to physical-domain geometry can be obtained, such as an approximation by a polygon.

My work in the past year has attempted to implement analytic methods to obtain the mapping function, in particular the Schwarz-Christoffel integral that maps the unit circle onto the interior of an n-sided polygonal domain, which is restricted to be convex. Riemann's mapping theorem applied to the Schwarz-Christoffel integral allows three of the n+z coefficients to be fixed arbitrarily. The program first estimates values for the remaining n-1 coefficients, determines an improvement, and continues until it attains a predetermined accuracy. At present, this interactive scheme is completely free of manual intervention, except in a few special cases which require much careful and tedious intervention.

I shall free the iteration scheme of manual intervention and make complex-plane numerical-integration techniques, especially in the vicinity of a pole, more accurate in the evaluation of the coefficients. A result of this increased accuracy will be an improvement in the iteration scheme, since one of its present failures is due in part to an accumulation of error in the determination of the length of the boundary.

Computer-Aided Teaching of Dynamic Systems Behavior - Ronald C. Rosenberg

The three main aspects of our work in the past year were: the continued development of the ENPORT system for use as a simulated dynamic-system laboratory, the extension of part of its programming algorithm to a hand procedure for student use in analysis, and the introduction of a simulated blank box as a significant teaching tool.

The ENPORT system can simulate the behavior of linear, lumped-parameter dynamic systems up to fifteenth order. Its input/output request forms have been simplified, but have been made more flexible by improved programming techniques. In addition, a local digital memory to the teletype-station-associated analog computer, which are automatically setup by the ENPORT program, provide a signal-versus-time oscilloscope display that represents the dynamic system behavior.

The analysis procedure starts with a bond graph model, assigns computing causality and a sign convention, and selects state variables. The system equations may then be written from the graph in an organized form and reduced to a standard form algorithmically. By means of a black box, a user can specify the input and observe the output in order to discover the unknown system. The system has been successfully operated as an on-line aid to classroom teaching in a class given by Prof. H. M. Paynter.

Automatic Network Synthesis - Marc R. Weinberger

In the past year I attempted to find good search strategies in a parameter space. A strategy achieves optimum performance through a choice of several possible network structures followed by a search for a best parameter combination. It is called optimum if it requires the fewest trials or the least time on a computer. I found a strategy which is optimum under certain conditions and often substandard in other cases where the required conditions are not met. Instead of conducting a single series of trials in the entire search space, this procedure conducts a multi-stage search in ever-narrowing subspaces. Simple rules indicate the allocation of the entire search to successive stages. This procedure leads to a large decrease in the number of trials and is more efficient than gradient-type methods or other deterministic strategies in several nontrivial cases.

I also studied the effect of noise in the measurements or calculations. Since the random search appears to be less affected by these errors than gradient-like methods in several instances, it has a broad field of applicability. The designer must be able to override the automatic search program, whenever he wants to try out his own ideas. Indeed, he learns more about his networks during the early stages of design. Since the randomness of the chosen trial points often yields good results from the very beginning, the designer may want to cut off the automatic routine in order to investigate more closely the excellent early returns. Thus, an interaction between man and machine, such as the one afforded by time-sharing, is a very important part of this method.

A Computer Model of Kinesthesis - George Piotrowski

The purpose of this work is to experiment with the use of the kinesthetic in communicating between man and computer. The model of kinesthesis will be a radially-extensible pivoting arm which has digital shaft encoders for position measurement and computer-controlled torque motors to provide a force output. The arm will cover a field of about fifteen inches square and be able to exert about five pounds of force. The first version of the arm will be as simple as possible and have most of the computational functions carried out by software instead of hardware. Thus far, I have written a program to simulate the device and also a control program for the device; and I am using the two to ascertain the effect of several design parameters, notably the sampling rate and the accuracy of measurement, on overall performance. With the control program, a user can select various mechanical systems for simulation. This program has been written in a general fashion, so that other algorithms can be added very easily.

Computer Solutions for Boundary-Value Problems - Coyt C. Tillman

I am developing a general, easy-to-use system for the solution of dimensional boundary-value problems. The programs written so far are a command processor, together with associated I/O and free-storage routines, boundary shape description routines, and a set of mathematical routines for the interpretation of arithmetic argument strings. The command processor, though written with boundary-value problems in mind, will be more generally usable. It admits to completely format-free input and, at the same time, avoids the imposition of the stringent punctuation conventions of other format-independent input interpreters.

A macro facility provides for the definition of new, composite commands, the modification of the system's vocabulary, and so on. I am now attending to the requirements of the data structure needed to represent the difference equations by which boundary value problems are being modelled.

Algebraic Expression Compiler - Grover C. Gregory

I am developing a compiler for algebraic expressions. It is primarily intended for use under the supervisory system that Coyt Tillman has described in this section.

The most recent work has been on the logic of the compiler's second pass. This pass is responsible for producing the appropriate machine code, using as its input the tables generated during the first pass. The logic is similar to that found in the MAD compiler which interprets lists to decide how to arrange the machine code. The aim has been to retain the flexibility of using such limits while reducing the amount of core they require. To that end, a different logic is used to handle the triple interactions. In addition, the compiler is constructed so that it sees variables and unsubscripted functions as identical, a requirement of the system under which it will operate. Testing of the whole compiler, as well as modifications to allow handling of subscripted variables and Boolean conditionals, will begin soon.

Algorithm to Aid Logic Design - Gerald A. Maley

My main interest at the beginning of this project lay in the development of a computer algorithm for the design of minimum Nor-Nand circuits from Boolean expressions. Algorithms do exist in this field for the limited case where complements of the input variables are available; but, in the design of an actual computer, these complemented variables are seldom available. When complements are available, only two levels of logic are required; but, in the general case, that of no complements, three levels are required. With the available computer time, I was able to develop the required algorithm for this type of implementation.

The proposing of an algorithm for a problem such as this is relatively simple, but the proving or disproving of such an algorithm can take months. On a remote console I was able to propose an algorithm and run several hundred cases in an hour or so. With response times of this order, I was able to find delinquencies and modify the algorithm in a matter of days, rather than months. Iterations of this kind finally produced a simple algorithm that I would not have uncovered with normal machine methods.

SCHOOL OF HUMANITIES AND SOCIAL SCIENCE

Social Systems Analysis

Academic Staff

G. Angell	E. Kuh	I. D. Pool

Non-Academic Research Staff

G. M. Bonham	N. I. Morris	S. Sacks
D. Levine	J. Nagle	T. H. Van Vleck
S. McIntosh	J. Rizika	

Research Assistants and other Students

P. Bos	D. A. Kendrick	S. Popkin
P. B. Clark	A. Kessler	L. Roos
R. Crossley	J. F. Kramer	H. L. Selesnick
C. Ellison	P. C. Ordershook	W. Selles
D. Griffel		

Social Systems Analysis - Ithiel D. Pool

We are currently working on a generalized editing system for social science data. Over past decades the handling of accumulated large files of survey records and census records have presented two major problems: the adaptation of older files to a computer, and the presence of errors in the data that will mislead a computer program. We are developing editing routines for on-line revision and correction of these older files. The editing system needs to compare data records with the protocol of allowable forms and provide feedback with respect to the editing decisions. The first version of the editing system is now being used, and, the experience of its users is very helpful in the current design phase of the next version.

The social science data surveys and census records on which we focus consists of both a codebook and records that describe individual people in ways outlined in the codebook. The editing system reorganizes the record, and a set of programs handles the codebook. The codebook requires that: 1) the data be put into computer readable form; 2) parts of the data be tagged by people to indicate characteristics of its form and content; and 3) the data be checked for internal discrepancies and the user informed of these on-line so that errors may be corrected.

In the CONCOM Project (ARPA Contract No. 920F-9717) we try to simulate the flow of messages in a nation. We first represent the audience by creating a population of hypothetical individuals that would constitute a fair sample of a real population. To describe our hypothetical population of individuals, we reconstructed from aggregate statistical data a possible distribution of facts about individuals compatible with the reported statistics. In this description we used a technique, developed by Frederick Mosteller, which synthesizes an n-dimensional table from any number of m-dimensional tables that constitute its marginals. For each of the cells into which the population is divided, this technique determines its frequency and several of its media characteristics. With it we can use not only marginals but also whatever estimates of interactions are available.

The next part of the simulation produces estimates of probable media exposure by the persons in the hypothetical population. An audience estimate is usually gotten by some kind of transformation of a circulation or set estimate; for example, the number of radio sets, the total daily newspaper circulation, the total magazine circulation, total book publication, and so on. All circulation figures then need to be multiplied by factors to provide audience estimates. In such total audience estimates, we use the Mosteller technique to take account of all marginals about which we have information and all interactions about which we have information.

To make our simulation dynamic, we must take into account differences in audience habits. For example, a newspaper read by one million people each day might conceivably be read by one million people 365 times a year or it might conceivably be read by 365 million different people only once. The actual distribution lies somewhere in between. We refer to this distribution of readers by frequency as the concept of cumulation. We have developed programs that estimate the underlying distribution of probabilities for any available cumulation data and generate cumulation curves from assumptions about the structure of the medium.

Our simulation also accounts for duplication: that is, the incidence of shared audience among different media. Exposure probabilities for each individual in the population are thus assigned on the basis of mean probabilities for the cell of the population to which he belongs, the likely shape of the probability distribution as indicated by cumulation facts, and the likely non-random duplication among media. In the final part of the simulation, a reader of a particular medium will note a particular news story in it with a probability p, which need not be constant. In the early period of a crisis, when the saliency of the events is low, p may be small; later, it may gradually rise. p may also vary with different groups in the population. With the simulation, we can thus formulate a set of assumptions about the flow of news in a total society and reach a conclusion as to who has been reached and how often. At present, we have a working model written in FORTRAN which we are running on test data and we are translating the entire set of programs into AED-O in order to run the simulation with on-line human decision making.

With CRISISCOM Project support from the CONCOM Project, we have developed a different siumlation of an aspect of crisis communications, namely the CRISISCOM siumlation (partially supported by the Naval Research Laboratory). Over the past year we have tested CRISISCOM by representing the Kaiser and the Tsar as they receive about 1500 messages covering events which occurred during the week before World War I. On-line, an experimenter manipulates the initial perceptions of each decision maker, the parameters of his selective perception biases, his distortions, and his forgetting mechanisms. He can examine the cognitive structure and attention spaces of the decision makers and display the spaces in a variety of ways. Allen Kessler is the main programmer of this model.

BLANK PAGE

SCHOOL OF SCIENCE

Time-Sharing vs. Ordinary Computer Usage

Electron Scattering Spectra

Bound-State Wave Functions

Nucleon-Nucleon Interactions

Magnet Design Parameters

Mathematical Research and CTSS

Academic Staff

L. N. Howard
A. H. Rosenfeld

Non-Academic Research Staff

E. C. Bartels
E. Campbell
B. Johanson
E. Miller
J. E. Spencer
N. C. Spencer

Research Assistant

D. D. Ryu

Time-sharing vs. Ordinary Computer Usage - Elizabeth J. Campbell

To compile actual statistics of CTSS compared with the usual method of batch processing is difficult, because the approach to the program is usually different. On the console, a programmer tends to be more adventuresome and try new approaches to a problem. The long turn-around time of batch processing causes a programmer to write a safe, conservative program. Comparison is also difficult because when a programmer uses batch processing, several programs are going at a time; while in CTSS, it is possible to concentrate on one program and save time, since a train of thought is not interrupted. The following examples illustrate this difference.

In one afternoon, an inexperienced programmer and I worked out a routine for Q_1, the Legendre polynomial of the 2^{nd} kind, to replace an inaccurate section of a program sent to us from Berkeley. We tried three new methods, comparing accuracy against tabulated results. This work would have taken us two to three days with batch-processing procedures.

In another instance, I spent about three weeks trying to build a Bound-State program from pieces of an old program. I decided not to do it on time-sharing because there would be many changes due to the differences between FORTRAN and MADTRAN. However, when the delay proved too great, I managed to convert the whole program to CTSS in a weekend. I decided at the time that I could have done the whole job on CTSS in a third to a half the batch-processing time.

Electron Scattering Spectra - Elaine Miller

I have been working on a set of programs that involve problems associated with the radiative degradation of electron scattering spectra. One of these programs is used as a source for predictions of data. These data are employed in a program that introduces all radiative effects into otherwise undegraded spectra. The program, when used as the source of predictions, computes elastic and inelastic yields throughout triangles selected according to the most reliable predictions. Both straight pole fit and Yang assumption are available for the computation of nucleon form factors.

Since, when primary electron energy is increased to 20 BeV, the yield often exists only within a very small portion of the selected triangle, this small portion is to be expanded into a larger area. Using time-sharing, I tested various methods of decreasing mesh size, thereby increasing area. Through comparison, elimination, and merging of methods, a new, efficient method evolved. For each set of data, time-sharing will be used to determine a new triangle, that strictly adheres to the conditions imposed upon the original triangle, but uses a finer mesh to obtain additional information.

<u>Bound-State Wave Functions</u> - Elmer Bartels

I have revised and perfected a program, which Elizabeth Campbell started with the Bound-State calculation, as the first of a package of programs for the use of physicists. A question-and-answer set of routines compute the normalized Bound-State wave function and binding energy for a Wood-Saxon well. The user may LOADGO the proper set of files and then type in well parameters and eigenvalue quantum numbers as they are requested. He has the option to guess the binding energy, or the program will supply the guess and subsequently compute the wave function and eigenvalue. Another option in the program stores from two to four Bound-State wave functions for computing the radial integral

$$I = \int_0^\infty R_1R_2R_3R_4dR.$$

The user may either compute another set of wave functions and binding energies with the same well parameters, or he may return to the beginning of the program and insert all new parameters of the Wood-Saxon well without having to LOADGO the files again.

Nucleon-Nucleon Interactions - Nancy Spencer

This use of CTSS is to investigate the applicability of the boundary condition model, developed by Professor Feshback and Dr. Lomon, to the proton-proton and neutron-proton scattering data in the energy range extending to 350 MeV. A basic set of parameters, necessary to specify the problem, are read into a program, which calculates such data as scattering length, partial cross section, total cross section, and polarization.

I then compare these theoretical results, predicted by the model, to experimental data at various energies and apply a chi-square criterion to determine goodness-of-fit. After the program is run off-line, each parameter is varied systematically by a specified amount until an optimum set of parameters appears.

Since only one parameter varies at a time, this process does not take into account any correlation which may exist between the parameters. Thus, with CTSS, a physicist can examine the results as they are computed and can interrupt the program and change a parameter whenever he sees a correlation or feels that one may exist. With such an interaction between a physicist and a computer, the development model, which gives a precision fit to the data, can proceed rapidly.

Magnet Design Parameters - James E. Spencer

The main purpose of this problem is to produce a computer program capable of calculating all first and second-order focusing coefficients for a serial array of magnets. The design parameters for these magnets are arbitrary, so that any geometrical form of boundary can be studied. Functional forms of greater than second order (for the geometrical form of boundaries) will not affect the first and second-order focusing coefficients, but they are included so the program can be used as a general ray-tracing subroutine. A realistic field description is being used, including fringing fields at entrance and exit boundaries which take into consideration the effects of finite widths and curvatures on the iso-induction lines.

Analytic expressions for the fringing fields in the median plane are obtained from experimental data by curve fitting. A knowledge of the field off the median plane is then obtained using Taylor expansions through the third order. The resulting expressions for the field are then used with equations of motion to trace rays of any initial conditions through the system.

Once the performance of this program has been determined by making specific calculations on known cases, other cases of interest will be investigated with a view towards learning the utility of these coefficients in the actual design of magnets.

Mathematical Research and CTSS - Louis N. Howard

This project has been largely directed toward exploring some of the potentialities of CTSS for assisting a mathematician in "everyday" aspects of his research. In general, most pure mathematicians, and many applied mathematicians, almost never use a computer. This is probably because computers have seemed most appropriate for large computations on completely-formulated and numerically-analyzed problems; and mathematicians are seldom interested in such problems. However, it seems that there are various special aspects of mathematical research in which the ability to rapidly make relatively-simple, tedious computations, not necessarily of a strictly numerical nature, could be of considerable use; for example in seeking counter-examples, testing conjectures, or trying to obtain an overall numerical or geometrical understanding of some formula or relationship. CTSS, used with certain basic programs designed to be fairly self-explanatory, seems to offer the possibility of providing such service to mathematicians without requiring much knowledge of programming.

We have experimented with several programs oriented toward this kind of work. One example is a very flexible and almost self-explanatory program for constructing rough typewriter graphs of contour lines for arbitrarily-given real functions of two-variables. This is quite easy for the uninitiated to use, and has been found valuable in obtaining a qualitative "feeling" for the behaviour of such functions. Another example is a program for making algebraic computations with matrices whose entries are elements of an arbitrary finite field. The field is specified by giving its characteristic p, number of elements, and an irreducible polynomial over the integers modulo p. A number of matrices can be handled simultaneously and operations with them are carried out by giving various simple commands. A third example is a program for constructing the Lagrange interpolation polynomial through an arbitrary set of points and carrying out various operations with this polynomial on the basis of simple commands. Other similar programs have been explored for computations in group theory, differential equations (specially adapted to time-sharing), and number theory. Finally, a program still under development is intended to provide a simple and flexible means for carrying out operations and sequences of operations that occur in elementary numerical analysis. If a suitable program can be developed, we hope to have students use it for a numerical analysis class.

While the greatest use of CTSS by mathematicians is likely to be with the type of small-scale everyday problems mentioned previously, some mathematicians will occasionally be interested in relatively-large-scale computational problems. Two such problems in which time-sharing appears to be particularly useful have been explored, partially at Project MAC and partially at the Computation Center. One is an elaborate program for studying the structure of finite groups, developed by Dr. C. C. Sims; the other, a set of programs for the study of problems in hydrodynamic stability, developed by L. N. Howard and Prof. M. Landahl of the Aeronautics Department. In both of these cases, the possibility of guiding computation on the basis of previous results has proved to be very useful.

In general, these experiments suggest that a time-sharing system can be effectively and usefully employed in some parts of mathematical research; though for general applicability, further simplification of programming, and simplifying at least a part of the command system of CTSS, should be brought about. The principal limitation of the specialized programs mentioned previously is that, in the case of complicated, flexible programs which actually do relatively little computation, like the contour-line program and the numerical-analysis program, the amount of time lost in swapping seems excessive.

SLOAN SCHOOL OF MANAGEMENT

On-Line Simulation System

The OPS-3 System

Computer Techniques in Business Problems

Console-Operated Statistical Routines

An Automated Stock Exchange

Computer-Planning System

Industrial Dynamics

On-Line Array Processor

Academic Staff

S. S. Alexander
A. E. Amstutz
D. C. Carroll
W. M. Evan
D. E. Farrar
E. P. C. Fernando
W. R. Fey
J. W. Forrester
M. Greenberger
J. D. C. Little
A. L. Pugh, III
E. B. Roberts
H. M. Weingartner

Non-Academic Research Staff

M. E. Wantman
S. H. Whitelaw
C. H. Willson

Research Assistants and other Students

P. Clermont
D. S. Diamond
J. Ever
G. C. Everest
L. Fattal
G. A. Gorry, Jr.
J. Huffman
T. J. Johnson
M. M. Jones
L. S. Kanodia
E. Kanstroom
J. L. Linderman
L. F. McPherson, III
J. R. Miller, III
J. H. Morris, Jr.
D. N. Ness
O. C. Nord
F. J. Russo
L. L. Selwyn
P. M. Wolk

On-Line Simulation System - Martin Greenberger and Malcolm M. Jones

The availability of time-sharing systems makes development of an on-line simulator both feasible and desirable. From our experience with GPSS, SIMSCRIPT, and a preliminary on-line simulator called OPSIM, we have drawn up specifications for a flexible, interactive simulation system. Our on-line simulator uses OPS, a multi-purpose system with compound operators (KOP). One KOP may call another and transmit parameters by way of a calling sequence; one may also activate or schedule the occurrence of another by means of special schedule operators. A special compound operator, called AGENDA, contains a list of future activities, which are automatically removed from AGENDA after it has been activated. A user may at any time inspect and modify AGENDA.

The execution of a compound operator may result in the alteration of state variables stored in the common system, or it may result in changes to AGENDA. Among the operators that change AGENDA are WAIT and DELAY WAIT is conditional upon the value of a state variable, and DELAY is unconditional and specifies that a certain period of time elapse before execution of the KOP is continued. Thus, activities listed in the AGENDA have either a time of execution or a Boolean condition associated with them. Two other operators allow activities scheduled in the AGENDA to be cancelled or rescheduled. In addition, the LOCAL operator lists variables to be saved when an activity is interrupted by DELAY or WAIT.

Among the simulations in progress or completed are:

1. A simple multi-server queuing model
2. An inventory distribution system
3. A process-control computer system
4. A three-level priority interrupt scheduling system (similar to the CTSS scheduling algorithm)
5. A hospital reception desk
6. An automated stock exchange.

The OPS-3 System - Martin Greenberger, James H. Morris, Jr., and David N. Ness

OPS-3 is a multi-purpose on-line system within which a user can write and test his own programs regardless of the nature of his particular project. Since the system is completely open, a user can add his own subroutines and use them in conjunction with system subroutines, as long as they obey customary conventions regarding common storage. He is free to run his programs within the system during the formative and testing stages, and then run them independently during the production stage.

The system contains operators for symbolic vector and matrix calculations, statistical analysis, live data-base reorganization, dynamic loading of routines, FORTRAN-like program writing with instant execution, and on-line simulations. While sitting at a console, the user may access information about the system. The information is stored in regular CTSS disc files and may be retrieved from within the OPS-3 system by a single key-word mechanism. OPS-3 is now being used in a number of projects. These include:

1. an on-line simulation system within which a user may change the course of the simulation or examine partial results,
2. a real-time simulation of the stock exchange that involves both simulated and console-initiated orders,
3. a polynomial manipulation system,
4. a system for statistical analysis in the social sciences,
5. a general debugging system for MAD programs in which programs are first run interpretively and then compiled into regular BSS subroutines.

OPS-3 is currently available as the CTEST6 command at Project MAC.

Computer Techniques in Business Problems - Donald C. Carroll, Paul Clermont, Thomas J. Johnson, Lalit Kanodia, and Francis J. Russo

This project investigates the use of heuristic programming and interactive problem-solving techniques in complex business problems, such as scheduling and sequencing production in a job shop. We developed a simulation model of a hypothetical shop, and, by testing some heuristics procedures for the sequencing and routing of jobs under varied and extensive conditions, determined the general steady-state behavior of this model. The first part of our research studied fully-automatic decision-making, in which the heuristics for sequencing and routing require no human guidance; the second part is to study the rule of a human pattern-recognizer and decision-maker who is closely coupled with the automatic system. In the latter research, we are investigating the establishment of manning levels for the shop, overtime decisions, and load-leveling. These decisions determine the structure under which an automatic decision system must operate; consequently, we call them structuring decisions.

A complete set of programs that study non-interactive decision making is now available. It includes a highly-parametric input generator (MADTRAN), a variety of simulation models that encompass such features as multiple-channel work stations, multiple-component orders, alternate-routing possibilities, and machine substitution (FAP), and finally an output analysis program (MADTRAN). Three pieces of research are complete. One, a study of heuristic sequencing rules for single and multiple components orders, demonstrated the general superiority of the so-called COVERT rule for sequencing. Another, which dealt with heuristics for selecting alternate routes for jobs, produced a superior heuristic derived from COVERT. The last study, on the effect of feed-back phenomena in heavily-loaded shops, led to several generalizations about the control of inputs and processing rates to obtain stable operation. A by-product of these studies of the minutiae of the decision system is a comprehensive understanding of the behavior of these models with respect to their aggregate behavior.

An extension of these studies to heuristics for machine substitution is almost complete; however, the major effort is in interactive decision-making. Toward this end, we are elaborating some of the basic programs to include such structural parameters as manning levels by shift, and we are developing generalized information retrieval and display programs for the ESL Display Console. A rudimentary version of our interactive system should be available in the fall.

Console-Operated Statistical Routines - James R. Miller and Stephen Whitelaw

A battery of statistical routines have been written, tested, and debugged for use with a CTSS console, primarily for the testing of hunches and limited hypotheses. Data are typed directly into the console, analyses are performed, and results are printed out immediately. Many of our routines are non-parametric in nature, and are oriented toward the analysis of typical problems in the social sciences. They include:

1. Standard two-dimensional contingency analyses.
2. Special three-dimensional contingency analyses.
3. Rank-order intercorrelation and partial correlation analyses.
4. Binomial tests.
5. Single and multiple percentage difference tests.
6. Homogeneity test on samples classified strictly according to nomial categories.
7. The Mann-Whitney U test of the difference of the sample medians.

We have also written several parametric routines:

1. Linear intercorrelation and partial correlation analyses.
2. Simple and multiple regression analyses.
3. A T-test of the difference between two sample means.
4. A one-way analysis of variance.
5. A test for normality of sample data.

We shall expand the range and capability of our current library by improving existing routines and by adding new ones. When S. Whitelaw has adapted the routines to the OPS system, a user will be able to call routines by name from the console, with the name of the data matrix as argument, to compound routines, and to save data and results for future use.

An Automated Stock Exchange - Stephen Whitelaw

With the OPS system, methods of automating large securities markets, such as the New York Stock Exchange have been explored. This research has included the programming of subroutines or operators to automatically handle all functions of a large stock exchange: entry of orders, consummation of orders, modification of completed transactions, and automatic bookkeeping. Additional operators embody the decision rules of specialists who maintain a market by committing capital.

Several sets of decision rules have been tested on different simulated markets. The performance of these rules is measured by price and net return criteria. The market model accepts simulated orders from an order generator. By treating a common file as an intermediate storage area for incoming orders, reports of completed transactions, requests for quotations, and output of information, it can also accept orders entered from other consoles. The system gives faster response times than those found in non-automated markets, and a programmed specialist is able to control the market under very adverse conditions.

Currently the system is being reprogrammed to function in two modes: mode 1 provides for continuous simulation and output of consummated transactions, and mode 2 for the entry of console orders into the simulation. Since the console remains undisturbed by this entry, it can function both as a ticker tape that reports the results of the simulated market, and also as a broker's device that enters orders into the same market. This functioning should make an effective management game and extend the use of the automated exchange as a research tool. The simulated market will be synchronized with the computer clock to prevent the simulation from getting ahead of real time.

Computer-Planning System - James C. Emery

This project studies a management computer planning system. Because planning requires a system that can be augmented and modified extensively, careful attention has been paid to general techniques that provide for such evolution. With such a model, a planner will be able to make sequential, hierarchical search for improved plans. First, from an aggregate model he will construct a large-scale plan, which is generated through a sequential search of alt rnatives. In this search, the human planner proposes alternatives and the computer transforms a proposed plan into its predicted consequences. Each probe for an improved plan provides the planner with additional information for the next probe. If the process converges, the aggregate plan emerging from it is then amplified by more detailed models within the constraints provided by large-scale plans. This hierarchical process continues until plans are sufficiently detailed for execution.

The system, specifically designed for budget planning, accepts a fairly wide class of functional relationship among variables. It will also deal with hierarchically-defined variables, such as manufacturing cost defined as the sum of direct labor, direct material, and overhead cost. Since functional relationships and the hierarchical definition of variables are represented primarily in the form of list structures, the system has the advantages of both flexibility and economy of storage. During the past year, I have developed the general structure of the system and have also begun to specify storage organization and the algorithms for the retrieval and manipulation of stored data and to define the set of subroutines needed for the generation of displays for budget planning.

Industrial Dynamics - Jay W. Forrester

The Industrial Dynamics Group uses time-sharing and the DYNAMO Compiler for analysis and design of feedback processes in industrial organizations. These systems contain multiple-loop feedback structures with highly-nonlinear relationships. Sixtieth-order models of systems have been simulated which contain several hundred variables. Such mathematically intractable systems contain important modes of behavior which are totally absent in the simpler linear systems. The DYNAMO Compiler is very efficient; for, in only a few seconds of computer time, it can simulate a system of a hundred or more equations through two hundred time steps. The Compiler contains a graphical plotting routine which makes time plots of systems responses on remote printing stations.

One use of the system has explored the dynamics of corporate growth. Many of the important growth processes depend on the interactions between positive feedback loops, negative feedback loops, and the coupling between these created by nonlinear elements. Another use has developed illustrative material for a student seminar on the principles underlying the behavior of feedback systems. These principles are generalizations about system behavior which have not previously been organized and illustrated. Some of them are implicit in the mathematics of linear feedback systems, but have not been clearly identified and illustrated; others exist only in multiple-loop nonlinear systems and could not have been previously identified without the extensive empirical results obtained by computer simulation.

On-line Array Processor - Mayer E. Wantman

Operators have been developed on the OPS-3 on-line system that provide a simple, flexible way of manipulating and analyzing numerical data within the computer. They can be used for general processing arrays, fitting of regression models, transformation of variables, rank-ordering of data, calculation of roots of polynomials of arbitrary degree, error analysis, generalized desk calculation, and MAD-type programming. They are:

1. Assignment Operator. The operator SET specifies and executes MAD-type assignment statements. Operands may be vectors and matrices, as well as elements. In addition, SET possesses embedded functions,

calls, matrix multiplication and transposition, a power generator, and a variety of logical operations. Lines are of the form

SET ALPHA=(BETA + GAMMA)*DELTA. P. (A + B)

2. Regression Operator. The operator FIT creates a multiple linear or simple polynomial model derived from given arguments. Sample points may be weighted and residuals saved. As an example, multiple linear regression of a dependent variable (stored in the vector named ALPHA) on independent variables (stored in the vectors named BETA, GAMMA, and DELTA) would be performed by the line

FIT ALPHA TO BETA GAMMA DELTA.

A cubic fit of ALPHA to BETA would be specified by

FIT ALPHA TO BETA DEGREE 3.

3. Root-Solving Operator. The operator ROOT uses the Newton-Raphson method of successive approximations for the determination of the root of a polynomial equation in one variable. Since the user specifies the starting value, he can obtain all roots by setting different initial values. Exponents need not be integral.

4. Ordering Operator. The operator RANK takes a vector of numbers and creates a second vector which is an ordered version of the first. A vector which describes the mapping performed may also be calculated.

5. Sorting Operator. The operator SORT sorts by intervals on a specified vector. The intervals may be specified either explicitly or by range and spacing. If any samples fall outside the sorting range, an appropriate message is printed.

APPENDICES

Appendix A

Project MAC Memoranda

Appendix B

M.I.T. Thesis

Appendix C

External Publication

Appendix D

Project MAC Technical Reports

APPENDIX A

PROJECT MAC MEMORANDA

MEMORANDUM MAC-M/No.	SUBJECT	AUTHOR	DATE
172	On-Line Documentation of the Compatible Time-Sharing System	J. Winett	7/21/64
173	SQZ/PADBIN (Ref: MAC-M-157)	R. Hussey	7/20/64
174	Abstracts of CTSS Console Commands (CC239) (174-2, 1/4/65; 174-3, 3/7/65	E. Kliman	8/3/64
175	Access to the MAC System via Alternate-Use-TWX ("TWX+Prime")	R. G. Mills	7/24/64
176	String Manipulation in the New Language (AIP memo 71)	D. Bobrow	7/28/64
177	CTSS Library Abstracts (CC240) (177-2, 11/23/64; 177-3, 3/22/65	E. Kliman	8/6/64
178	Crunch Command in CTSS	R. Bayles	8/7/64
179	BEFAP Command within CTSS	R. Bayles	8/10/64
180	The ARCHIV Command	A. Scherr	10/19/64
181	The Structure of On-Line Information Processing Systems	E. Glaser J. Dennis	10/1/64
182	Preliminary Notes on the Hard Matter for the MAC 635	E. Glaser	8/28/64
183	Proposed Instructions on the GE 635 for List Processing and Push Down Stacks (AIP memo 72)	M. Levin	9/18/64
184	On Memory Addressing in the GE 635	M. Minsky M. Levin	9/22/64
185	MAC and Its Users	U. Neisser	9/29/64
186	A Note on Asynchronous Parallel Processing (MSG memo 5)	H. Witsenhausen	7/64
187	Nesting and Recursion of Procedure in Segmented Memory (MSG memo 6)	J. Dennis E. C. Van Horn	10/28/64
188	Automatic Scheduling of Priority Processes	J. Dennis	10/64

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
189	An Example of Intersphere Communication and Asynchronous Parallel Processing: Typewriter Console Message Handling by Protected Service Routines (MSG memo 8)	J. Dennis	9/25/64
190	General Comments about PDP-6 Coding and Hardware (PDP-6 memo 1)	D. Edwards	9/25/64
191	TECO 6 (PDP-6 memo 2)	D. Edwards	
192	Unrecognizable Sets of Numbers (AIP memo 73)	M. Minsky S. Papert	11/2/64
193-2	TYPSET and RUNOFF, Memorandum Editor and Type-out Commands (CC 244-2)	J. Saltzer	11/6/64
194	On the Control Specifications of Computation	D. Kuck	11/64
195	ED, A Context Editor for Card Image Files (195-1, 3/15/65) (CC245; CC234-1)	R. Dailey	11/20/64
196	Some Aspects of the Structure of Computation	D. Kuck R. Kain	11/13/64
197	Some Time-Sharing Ideas	D. Kuck	11/13/64
198	PLEX-Dump and Relocation in the AED-O System (9442-M-110)	T. Fox	10/26/64
199	Stack Manipulation Facilities of the AED-O Language (9442-M-112)	R. Coe	11/18/64
200	Processes, Spheres of Protection and Independent Computation (MSG 9)	E. Van Horn	(no date)
201	New Operating System for the ESL Display Console (9443-M-118 (Cont'd 12/10/64)	R. Bayles	11/23/64
202	Proposal to Improve the Rotation Matrix of the ESL Display Console (9442-M-114)	R. Stotz	11/27/64
203	A Compacting Free Storage Program (9442-M-113	B. Wolman	3/5/65
204	AED-O Compiler for Batch Processing (9442-M-115	H. Spencer	12/2/64

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
205	MADBUG: A MAD Debugging System (CC247)	R. Fabry	11/25/64
206	CTSS LISP Notice: Supplement to A. I. Memo No. 67 (AIP memo 74)	T. Hart	12/7/64
207	Internal Memos for AED Users (9442-M-116)	C. Feldmann	12/8/64
208	AED Flash 10, New CTEST2 Command (9442-M-119)	C. Feldmann	12/10/64
209	The Naval Mine Warfare Game	I. Sutherland	12/18/64
210	Graphical Output Device	A. Rutchka	12/21/64
211	New TWX-Prime Lines	R.G. Mills	12/23/64
212*	Translation of Communication with Automata by Carl Adam Petri	O. Selfridge	6/10/65
213	AED FLASH 11: AED BUG Usage (9442-M-121)	T. Fox	12/28/64
215	Television Camera-to-Computer Adapter: PDP-6 Device 770 (AIP memo 75)	M. Minsky	1/9/65
216	New B-Core System for Programming the ESL Display Console (9442-M-122)	C. Lang	4/30/65
217	Operating Manual for the ESL Display Console (9442-M-129)	R. Stotz J. Ward	3/9/65
218	Social Science Data Bank Development	I. D. Poole	1/21/65
219	The COMIT Feature in LISP II (AIP memo 76) (LISP II Project Memo 2)	D. Bobrow	2/18/65
220	The COGO Language	D. Roos	2/25/65
221	The INFO Command	J. Winett	2/25/65
222	AED FLASH 12: Multi-Entry Procedures (9442-M-124)	D. Ross	2/25/65
224	An Automatic Plotting Subroutine, XYPLOT	T. Stockham, Jr.	3/3/65

* In older listings of MAC-M-memos, this memo was numbered as MAC-M-214.

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
225	AED FLASH 13: Argument Checking Procedures for AED (9442-M-125) (225-2, 4/15/65)	J. Walsh	3/5/65
226	AED FLASH 14: Availability of AED JR Systems (9442-M-126)	D. Ross	3/9/65
227	The Editing Routine	D. Griffel	3/8/65
228	A General Cross-Tabulation System (COM/COM Sim. memo 23) (See Additional Information on Cross-Tabs: New Output)	W. Selles	4/24/65
229	Proposed Datanet-30 Program	S. Dunten	3/18/65
230	Mind, Matter and Models (AIP memo 77)	M. Minsky	3/26/65
231	Description of SNOBOL Commands (CC235-2)	L. Pouzin	4/9/65
232	Preliminary Description of DAEMON, the Disk Dump and Reload Program Package, and its Operation (CC 252)	M. Bailey	4/26/65
233	Trial Launching of the OPS-3 System	M. Jones D. Ness	4/26/65
234	Users' Manual on the Use of Structural Language	R. Logcher	9/15/65
235	BLODI Command	O. Wright, Jr.	5/4/65
236	AED FLASH 15: Execution Time Procedure Calls (9442-M-131)	J. Walsh	4/22/65
237	A Generalized File Structure and Input/Output System (CC 241-1)	R. C. Daley R. J. Creasy R. M. Graham	5/7/65
238	Some Facilities of the MAC Document Room	Document Room (A. Bowen)	5/10/65
239	KREADQ, SETRDQ, MODRDQ, A Set of Format-Free Input Subroutines	D. Ness	6/16/65
240	Topics in Model Theory	M. Levin	5/25/65
241	PDP-6 LISP Input-Output for the Dataphone (AIP memo 79)	W. Martin	6/1/65

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
242	PDP-6 LISP Input-Output for the Display (AIP memo 80)	W. Martin	6/28/65
243	Specifications for a Dataphone-Driven Remote Display Console for Project MAC	R. H. Stotz J. E. Ward U. F. Gronemann	6/24/65
244	Proposal for Commands for Dynamic Resource Allocation (CC 254)	G. Schroeder	4/17/65
245	DIRECT; A Command Subroutine for Controlling Programs	T. Crystal E. River	6/23/65
246	Computer Experiments in Finite Algebra	W. D. Maurer	6/14/65
247	AED FLASH 16: Use of AED JR to Parse FORTRAN II (9442-M-134)	H. Spencer	6/25/65
248	Use of MACDMP (AIP memo 83)	P. Samson	7/1/65
249	MAC PDP-6 DECtape File Structure (AIP memo 82)	P. Samson	7/1/65
250	PDP-6 TECO, July 1965 Version	P. Samson	7/23/65
251	AED FLASH 17: Features of the Revised AED-JR (9442-M-136)	J. F. Walsh R. Feldmann	7/1/65
252	AED FLASH 18: A Subroutine for Generating Coons' Surfaces (9442-M-137)	C. Lang	6/30/65
253	FMS Tape Distribution (9442-M-127-3)	R. Feldmann D. E. Walker	7/1/65
254	Derivation of a Mean Pseudo-Associative Search Time (MSG memo 13)	E. C. Van Horn	8/16/65
255	Surfaces for Computer-Aided Design of Space Figures (9442-M-139)	S. A. Coons	7/21/65
256	Three-Dimension Pseudo Pen Sub-routine for use with the ESL Display Console (9442-M-138)	R. Polansky	7/13/65
257	Syntax and Display of Mathematical Expressions (AIP memo 85)	W. Martin	7/29/65

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
	. . . REVISIONS . . .		
168-2	A Subroutine Trace Program for CTSS	B. Wolman	3/5/65
174-3	CTSS Console Commands	E. Kliman	3/7/65
177-3	CTSS Library Subroutines	E. Kliman	3/22/65
193-2	TYPSET and RUNOFF, Memorandum Editor and Type-out Commands (CC-244-2)	J. Saltzer	1/11/65
195-1	ED, a Context Editor for FAP and MADTRN Program	R. Dailey	3/15/65

APPENDIX B

M. I. T. THESIS

Amstutz, A. E., A Management Oriented Behavorial Theory of Interactions within Consumer Product Markets, M. I. T. Sloan School of Management, Ph. D. Thesis, June 1965

Armstrong, A. E., A Braille Telecommunication Terminal, M. I. T. Department of Mechanical Engineering, S. M. Thesis, June 1965

Beckreck, L. N., SEPOL: A Soil Engineering Problem-Oriented Language, Department of Civil Engineering, M. S. Thesis, June 1965

Beusch, J. U., Dynamic Behavior and Control of Communication Networks, M. I. T., Department of Electrical Engineering, Ph. D. Thesis, May 1965

Brach, J. R., A Problem-Oriented Language for Project Management, M. I. T., Department of Civil Engineering, S. M. Thesis, January 1965

Carroll, D. C., Heuristic Sequencing of Single and Multiple Compound Jobs, M. I. T., Sloan School of Management, Ph. D. Thesis, June 1965

Chang, W. Y., Instability Analysis of Transverse Waves in Solid-State Plasmas, M. I. T., Department of Electrical Engineering, B. S. Thesis, June 1965

Chiodi, W., A Mechanism for the Transformation of S, M. I. T., Department of Electrical Engineering, S.M. Thesis, June 1964

Diamond, D. S., A Step toward Computer Aided Preparation of Magazine Advertising Copy, M. I. T. Sloan School of Management, B. S. Thesis, June 1965

Dvorak, A. Jr., An Input-Output Program for Electronic Circuits Using a CRT, M. I. T., Department of Electrical Engineering, Thesis, June 1965

Efimba, R. E., Analysis of a Bridge Deck as a Plane Grid, M. I. T., Department of Civil Engineering, M. S. Thesis, June 1965

Ellis, H. E., A MAD-based Programming Language, M. I. T., Department of Electrical Engineering, B. S. Thesis, June 1965

Falk, G., An Information Retrieval System for Reference Filing, M. I. T., Department of Electrical Engineering, B. S. Thesis, June 1965

Finkelstein, S. C., An Analysis of a Space Beam, M. I. T., Department of Civil Engineering, M. S. Thesis, June 1964

Fiske, R. S., A Real-Time System for Computer Monitoring of the Stock Exchange, M. I. T., Sloan School of Management, B. S. Thesis, June 1965

Foster, A. T., Special Input-Output Facilities for the IBM 1620, M. I. T., Department of Civil Engineering, M. S. Thesis, May 1964

Goeke, R. F., Introduction to Machine Language Programming on a Time-Shared IBM 7094, M. I. T., Department of Electrical Engineering, B. S. Thesis, June 1965

González-Zubieta, R., On Some Aspects of Integer Linear Programming, Ph. D. Thesis, Sloan School of Management, June 1965 (Also released by Operations Research Center as Technical Report No. 16, June 1965)

Goodman, R. V., A Preliminary Method of Analysis for a Computer-Aided Structural Design Language, M. I. T., Department of Civil Engineering, S. M. Thesis, June 1965

Groneman, U. F., Coding Color Pictures, M. I. T., Department of Electrical Engineering, Ph. D. Thesis, February 1964

Guttmann, E. G., Investigation of Threshold Logic in High-Speed Display Systems, M. I. T., Department of Electrical Engineering, S. M. Thesis, June 1965

Guzman Barron-Torres, G., Optimum Design of Prestressed Concrete Highway Bridges using Digital Computers, M. I. T., Department of Civil Engineering, S.M. Thesis, May 1965

Hadley, R. A., Investigation of Automated Direct Design of Rigid Steel Planar Frames, M. I. T., Department of Civil Engineering, M. S. Thesis, February 1965

Haines, E. C. Jr., The TREET List-Processing Language, M. I. T., Department of Electrical Engineering, M. S. Thesis, September 1964 (Also produced by Mitre Corporation as Report SR-133, April 1965)

Hamann, W. C., Computer Aided Design of Slender Structural Members, M. I. T., Department of Mechanical Engineering, M. S. Thesis, May 1964

Hamilton, M. L., and A. D. Weiss, An Approach to Computer-Aided Preliminary Design, M. I. T., Department of Mechanical Engineering and Naval Architecture, S.M. Thesis, August 1964 (Also as ESL-TM-228, January 1965)

Hershdorfer, A. M., Optimal Routing of Urban Traffic, M. I. T., Department of Civil Engineering, Ph.D. Thesis, June 1965

Hoag, J. D., Instabilities in Transverse Waves Along β for a Beam Type Distribution, M. I. T., Department of Electrical Engineering, B.S. Thesis, October 1964

Ishii, H., Instabilities in InSb, M. I. T., Department of Electrical Engineering, S.M. Thesis, May 1965

Johnson, W. F., Use of Traffic Volume Data in Evaluation of Highway User Costs, M. I. T., Department of Civil Engineering, S.M. Thesis, June 1964

Kanodia, L. S., Queue Balancing - Another Approach to the Theory of Scheduling, M. I. T. Sloan School of Management, M. S. Thesis, June 1965

Kanstroom, E., An Information Retrieval Program Compatible with the M. I. T. Time-Shared Computer System, M. I. T. Sloan School of Management, B.S. Thesis, June 1965

Kawaguchi, S. K., Preliminary Computation and Data Reduction for STRESS, M. I. T., Department of Civil Engineering, S.M. Thesis, August 1964

Keller, J., The Critical Path Method, Its Implementation and Effective Utilization in Construction, M. I. T., Department of Civil Engineering, M.S. Thesis, June 1964

Kim, Numerical Analysis of Open-Channel Transients from Hydro-Power Operation, M. I. T., Department of Civil Engineering, S.M. Thesis, February 1965

Kusse, B., Plasma Dispersion Relations and the Stability Criteria, M. I. T. Department of Electrical Engineering, S.M. Thesis, September 1964

Ladd, H. E., A Computer System for Classification, Storage and Retrieval of Design Information, M. I. T., Department of Mechanical Engineering, M.S. Thesis, January 1964

Lasher, D. A., A Multi-Processor, Multi-Memory Computer Simulator, M. I. T., Department of Electrical Engineering, B.S. Thesis, June 1965

Linder, W., Mathematical Object Description Language, M. I. T., Department of Civil Engineering, S.M. Thesis, January 1965

Luconi, F. L., Real-Time Braille Translation System, M. I. T., Department of Electrical Engineering, S.M. Thesis, June 1965

Males, R. M., A Dynamic Programming Model for Reservoir Simulation, M. I. T., Department of Civil Engineering, S.B. Thesis, August 1964

Mazzota, S. G., Method of Specifying Design Requirements within a Structural Design Language, M. I. T., Department of Civil Engineering, M.S. Thesis, June 1965

McNamara, J. E., Traffic and Trunking Requirements in a Data Transmission System, M. I. T., Department of Electrical Engineering, B.S. Thesis, June 1964

Mills, J. D., A Computer Display System for Analysis of Polynomial-Type Dispersion Relations, M. I. T., Department of Electrical Engineering, S.M. Thesis, February 1965

Morris, N. I., A Data Structure Programming Language, M. I. T., Department of Electrical Engineering, B.S. Thesis, June 1965

Murray, H. G., Computer Implementation of a Technique for the Transformation of Waveforms, M. I. T., Department of Electrical Engineering, S.B. Thesis, June 1965

Nanavati, H. M., Graphical Representation of Energetic Systems of Sketchpad, M. I. T., Department of Electrical Engineering, M. S. Thesis, June 1964

Patel, N. R., A Mathematical Analysis of Computer Time-Sharing Systems, M. I. T., Department of Electrical Engineering, M. S. Thesis, June 1964

Potter, R. L., The Optimal Allocation of System Resources in a Time-Shared Computer, M. I. T., Department of Electrical Engineering, S.M. Thesis, June 1965

Russo, F., A Heuristic Approach to Alternate Routing in a Job Shop, M. I. T., Sloan School of Management, S.M. Thesis, June 1965 (Brooks Prize in Industrial Management for the best thesis submitted for degree of M. S. in Industrial Management)

Santos, P., CADD, A Computer-Aided Digital Design Program, M. I. T., Department of Electrical Engineering, S.M. Thesis, June 1965

Scherr, A. L., An Analysis of Time-Shared Computer Systems, M. I. T., Department of Electrical Engineering, Ph. D. Thesis, June 1965

Selesnick, H., Overlapping Group Memberships, Cross-Pressures, and Cognitive Dissonance, M. I. T., Sloan School of Management, M. S. Thesis, June 1964

Sklar, J. R., Sequential Measurement of Multi-Dimensional Transducers, M. I. T., Department of Electrical Engineering, Ph. D. Thesis, September 1964 (Also Lincoln Laboratory, Report TR-360, October 1965)

Spitz, R., A Computer Evaluation of the Point and Figure Method of Stock Market Trading, M. I. T., Sloan School of Management, S. M. Thesis September 1964

Teague, L. C., Jr., Data Control Procedures in a Time-Sharing System for Structural Design, M. I. T., Department of Civil Engineering, M. S. Thesis, June 1965

Traub, A. C., Jr., Sensitivity Analysis of a Generalized Model of Consumer Behavior, M. I. T., Sloan School of Management, B. S. Thesis, June 1965

Wallace, R. N., An Investigation of Complex Waves in Electron Beam-Plasma Systems, M. I. T., Department of Electrical Engineering, S. M. Thesis, September 1964

Weiner, J. I., An Analysis of Graphical Input Systems for Computer-Aided Design Installations, Department of Mechanical Engineering, S. B. Thesis, May 1964

Welch, R. L., The Development of an On-Line Computer Simulation System, M. I. T., Sloan School of Management, S. M. Thesis, June 1964

Westerfeld, E. C., Flowchart Compiler Using Teager Board Input, M. I. T., Department of Electrical Engineering, M. S. Thesis, June 1965

Whitelaw, S. H., An Automated Stock Exchange, M. I. T., Sloan School of Management, S. M. Thesis, June 1965

Winett, J. M., On-Line Documentation of the Compatible Time-Sharing System, M. I. T., Department of Electrical Engineering, E. E. Thesis, February 1965 (Also Lincoln Laboratory Report No. 387, May 1965)

Wolk, P. M., A Program for Solving the Mixed Integer Linear Programming Problem, M. I. T., Sloan School of Management, B. S. Thesis, June 1965

Wolman, B. L., Operators for Manipulating Verbal and Graphical Language Structures, M. I. T., Department of Electrical Engineering, S. M. Thesis, June 1965

Yudkin, H. L., Channel State Testing in Information Decoding, M. I. T., Department of Electrical Engineering, Ph. D. Thesis, September 1964

Zeigler, B. P., Human Use of Short Term Memory in Processing Information on a Console, M. I. T., Department of Electrical Engineering, M. S. Thesis, June 1964

APPENDIX C

EXTERNAL PUBLICATION

Bers, A., "The Nature of Stable and Unstable Waves in Plasmas and Other Dispersive Media", Bulletin of the American Physics Society, Vol. 9, p. 557, June 1964

Bers, A., "Stability Criteria and Dispersion Relations", M. I. T. R. L. E. Quarterly Progress Report No. 76, January 1965

Bers, A., "Further Analysis of Interactions of an Electron Beam with Ions in a Warm Plasma", M.I.T. R.L.E. Quarterly Progress Report No. 76, January 1965

Bers, A., "Instabilities in Plasmas with Beam-Type Distribution", M. I. T. R. L. E. Quarterly Progress Report No. 76, January 1965; also in the Bulletin of the American Physics Society, Vol. 10, p. 221, February 1965

Bers, A., J. K. Hoag, and E. A. Robertson, "Instabilities in Transverse Waves along B_o for Beam-Type Distribution", M.I.T. R.L.E. Quarterly Progress Report No. 77, April 1965

Bers, A., S. Puri, and J. D. Mills, "Interaction of an Electron Beam with Ions in a Warm Plasma of Finite Transverse Dimensions", M.I.T. R.L.E. Quarterly Progress Report No. 74, J ly 1964

Bers, A., and S. Gruber, "Negative Energy Plasma Waves and Instabilities at Cyclotron Harmonics", Applied Physics Letters, Vol. 27, No. 6, January 1965

Bobrow, D. G. and J. Weizenbaum, "List Processing and Extension of Language Facility by Embedding", IEEE Transactions on Electronic Computers, August 1964

Bobrow, D. G., "A Question-Answering System for High School Algebra Word Problems", Proceedings of the Fall Joint Computer Conference, November 1964

Clarkson, G., "On the Theory of Price Behavior", M.I.T. Sloan School of Management Working Paper 107-65.

Clarkson, G., and F. P. Tuggle, "A Theory of Group Decision Behavior", M.I.T. Sloan School of Management Working Paper 92-64, September 1964

Corbató, F. J. (Chairman), "A Panel Discussion on Time-Sharing", Datamation, August 1964

Cozzolino, J. M., Jr., R. Gonzalez-Zubieta, and R. L. Miller, "Markovian Decision Processes with Uncertain Transition Probabilities", Operations Research Center Technical Report No. 11, March 1965

Crystal, T. H., "A Model of Larynx Activity during Phonation", M.I.T. R.L.E. Quarterly Progress Report No. 78, July 1965

Darlington, J. L., "Machine Methods for Proving Logical Arguments Expressed in English", IFIPS Congress 1965, May 1965

Dennis, J. B., "Group Report Computer Research", M.I.T. R.L.E. Quarterly Progress Report No. 76, January 1965

Dennis, J. B., "A Multi-User Computation Facility for Education and Research", Communications of the ACM, Vol. 7, No. 9, September 1964

Dennis, J. B., "Use of Computers in Speech Research", and "Computers in Medicine and Biology", Annual Report of the New York Academy of Science, Vol. 115, Art. 2, P. 867, July 1964

Dennis, J. B., "Segmentation and the Design of Multi-Programmed Computer Systems", IEEE International Convention Record, 1965, Part III, Proceedings, March 1965

Dennis, J. B., and E. L. Glaser, "Structure of On-Line Information Processing Systems", Second Congress on the Information System Sciences, November 1964

Dertouzos, M. L., "Threshold Element Synthesis", Report No. ESL-R-200, Electronic Systems Laboratory, M. I. T., June 1964

Emery, J. C., "The Impact of Information Technology on Organization", Proceedings of the 1964 Annual Meeting of the Academy of Management, Chicago, December 1964

Emery, J. C., "Planning as an Iterative Hierarchical Process and its Formalization in Computer Models", Second Congress of Information Systems Sciences, Spartan Press, November 1964

Ever, J., "A Corporate Time-Shared Computing System", Industrial Management Review, Spring, 1965

Fabry, R. S., "A COMIT Generating and Parsing Program for Left to Right Subscript Grammars with Discontinuous Constituents", IFIPS Congress Proceedings, Vol. 2, May 1965

Fano, R. M., "The MAC System: The Computer Utility Approach", IEEE Spectrum, January 1965, pp. 56-64 (See also MAC-TR-12.)

Fano, R. M., "Progress Report on Project MAC", Proceedings of the Ninth College and University Machine Recordings Conference, Monograph 4, April 1964

Feldmann, C. G., "Programming in AED-O", Spring APT Technical Meeting, Chicago, April 1965

Fenves, S. J., R. D. Logcher, and D. F. Reinschmidt, STRESS: A Problem-Oriented Language for Structural Engineering, M. I. T. Press, 1965

Foster, A. and R. Walker, Curve and Circle Subroutines for the CESL 1620 Plotter, M. I. T. Civil Engineering Systems Laboratory, March 1965

Foster, A. T., J. W. Weber, and R. A. Walter, FORTRAN Subroutines for Using the CESL 1620 as a CTSS Terminal, M. I. T. Civil Engineering Systems Laboratory, April 1965

Glaser, E. L. and F. J. Corbató, "Introduction to Time-Sharing", Datamation, Vol, 10, No. 11, November 1964

Greenberger, M., "A New Methodology for Computer Simulation", M.I.T. Sloan School of Management Working Paper 101-64, (Presented at the Conference on Computer Methods in the Analysis of Large-Scale Social Systems, Sponsored by the Joint Center for Urban Studies, October 1964

Greenberger, M., "Methods in Randomness", Communications of the ACM, Vol. 8, No. 3, March 1965

Greenberger, M., "Decline and Fall of the Check" (Banking and the Information Utility), Bankers Magazine, Summer 1965; also in Computers and Automation, Vol. 14, No. 4, April 1965

Greenberger, M., "A Multi-Purpose System for the Structuring On-Line of Information and Decision Processing", IFIPS Congress, Vol 11, Spartan Books, May 1965

Greenberger, M., Symposium on "Methods of Describing Information Systems", IFIPS Congress, Spartan Books, May 1965

Greenberger, M., "The Two Sides of Time-Sharing", IFIPS Congress 1965, Vol. 11, Spartan Books, May 1965

Greenberger, M., "Priority Scheduling of Time-Shared Computer Systems", Bulletin of Operations Research Society of America, Vol. 13, Supplement 1, Spring 1965

Hansen, K. F., and I. C. Pyle, "TREC; A Time-Sharing Reactor Code System", American Nuclear Society Transaction, Vol. 7, No. 2, November 1964

Hershdorfer, A. M., "Optimal Route Assignment for Metropolitan Road Traffic", Bulletin of the Operations Research Society of America, Vol. 13, Supplement 1, June 1965

Hershdorfer, A. M., and J. C. Frazier, User's Manual for the M. I. T. Time-Sharing Linear Programming System, M. I. T., Department of Civil Engineering, July 1964

Kessler, M. M., The M. I. T. Technical Information Project, Part 1, System Description, M. I. T., November 1964 (AD-608-502)

Kessler, M. M., E. L. Ivie and W. D. Mathews, "The M. I. T. Technical Information Project: A Prototype System", Proceedings of the American Documentation Institute, 1964

Kessler, M. M., "The M. I. T. Technical Information Project", Physics Today, March 1965

Kessler, M. M., "The History and Prospectus of TIP", Library Report S-3, April 1965

Kessler, A. R., and I. Pool, "CRISISCOM: A Computer Simulation of Perception and Decision-making during a Crisis", IEEE. International Convention Record, Part 7, Human Factors, March 1965

Lang, C. A., and R. B. Polansky, "Graphical Language, A Step towards Computer-Aided Design", IEEE Machine Tools Industry Conference, November 1964

Lieberman, M. A., "Dispersion Diagram for Hot-Electron Plasmas", M. I. T. RLE Quarterly Progress Report, No. 77, April 1965

Little, J. D. C., "A Model of Adaptive Control of Promotional Spending", M. I. T. Sloan School of Management Working Paper 122-65, May 1965

Llewellyn-Jones, D. T., "Use of Interferometric Spectroscopy for Far-Infrared Plasma Diagnostics", M. I. T. R. L. E. Quarterly Progress Report No. 74, July 1964

Llewellyn-Jones, D. T., and S. C. Brown, "Measurement of the Dielectric Coefficients of Dense Plasmas at Submillimeter Wavelengths", M. I. T. R. L. E. Quarterly Progress Report No. 76, January 1965

Logcher, R. D., "Form and Use of an Interactive Problem-Oriented Language", Fall Joint Computer Conference, November 1965

Maurer, W., Computer Experiments in Finite Groups, Brown University Research Center for Dynamic Systems, January 1965

Miller, C. L., and R. A. Walter, "Communicating with Computers in Civil Engineering Design", M. I. T. Civil Engineering Systems Laboratory, Technical Report T65-4, March 1965

Mills, J. D., "Computer Display System for Analyzing Instabilities", M. I. T. R. L. E. Quarterly Progress Report, No. 75, October 1964

Mills, R. G., "Communications Implications of the Project MAC Multiple-access Computer Systems", IEEE International Convention Record, Part I, March 1965

Minsky, M. L., "Matter, Minds, and Models", IFIPS Congress 1965, New York, Spartan Books, May 1965

Morey, J. L., and D. B. Yntema, "Experiments on Systems", Proceedings of the Second Congress on Information Systems Sciences, Spartan Books, November 1964

Nolan, J. F., and A. W. Armenti, "An Experimental On-Line Storage and Retrieval System", Lincoln Laboratory Report TR-377, February 1965

Peterson, H. A., and W. F. Higgins, "Computer Program for the Optimization of the Design of a Unified Carrier", Lincoln Laboratory Project Report AP-43 (Apollo), March 1965

Pitidis-Poutous, T., "The Mathematical Derivation of Preliminary Ship Lines", M. I. T. Department of Naval Architecture and Marine Engineering Report 65-3, February 1965

Pool, I. D., "Attitudinal Variables in Large-Scale Social Systems", Conference on Computer Methods in the Analysis of Large-Scale Social Systems, Joint Center for Urban Studies, October 1964

Pool, I. D., and A. Kessler, "The Kaiser, The Tsar, and The Computer: Information Processing in a Crisis", American Behavioral Scientist, Vol. 8, No. 9, May 1965

Pool, I. D., R. P. Abelson, and S. L. Popkin, "A Postscript on the 1964 Election", The American Behavioral Scientist, Vol. 8, No. 9, May 1965

Pyle, I. C., "Data Input by Question and Answer", Communications of the ACM, Vol. 8, No. 4, April 1965

Raphael, B., "A Computer which Understands", Fall Joint Computer Conference 1964, November 1964

Reinschmidt, K. F., "DATALINK, An On-Line Computing System for Structural Laboratory Research", M.I.T. Department of Civil Engineering Report R65-6, March 1965

Roberts, L. G., "Homogeneous Matrix Representation and Manipulation of N-Dimensional Constructs", University of Michigan Engineering Summer Conference, June 1965

Roos, D., and B. Schumacker, "ICES: Integrated Civil Engineering System", M. I. T. Civil Engineering Department Conference on Improved Highway Engineering Productivity, May 1965

Ross, D. T., AED Jr: An Experimental Language Processor, ESL-TM-211, M. I. T., September 1964

Ross, D. T., S. A. Coons, and J. E. Ward, Investigations in Computer-Aided Design for Numerically-Controlled Production, ESL Report S-359, M. I. T., June 1965

Ross, D. T., "Computer-Aided Design", 29th Annual Machine Tool Electrification Forum (Sponsored by Westinghouse), Buffalo, New York, April 1965

Ross, D. T., "System Provisions for the Debugging of AED Language Programs", IFIPS Congress 1965, New York, Spartan Books, May 1965

Ross, D. T., "Current Status of AED", Second Annual SHARE Design Automation Workshop, Atlantic City, New Jersey, June 1965

Selfridge, O. G., "Social Responsibility and Computers", Conference on Computer Methods in the Analysis of Large-Scale Social Systems, Joint Center for Urban Studies, October 1964

Selfridge, O. G., "Models of Learning", American Psychiatric Association Meeting, New York, May 1965

Selfridge, O. G., "Reasoning in Game-Playing by Machine", Symposium on Computer Augmentation of Human Reasoning, Spartan Press, 1965

Spangler, P. S., N. C. Rasmussen, and D. J. Rose, "Fusion Reactor Blanket Experiment", American Physical Society Meeting, November 1964

Stockham, T. G., Jr., "Some Methods of Graphical Debugging", IBM Man-Machine Communications Symposium, May 1965

Teager, H., "Workshop on Problem Solving through the Use of Automatic Displays", Fall Joint Computer Conference, November 1964

Teitelman, W., "Real-Time Recognition of Hand-Drawn Characters", Fall Joint Computer Conference, November 1964

Wallace, R. N., A. Bers, and S. Puri, "Electron-Beam Interaction with Ions in a Hot Electron Plasma", Bulletin of the American Physics Society, 10-222, February 1965

Weber, J. W., The Use of a Small Computer as a Remote Terminal for a Large-Scale Time-Shared Computer, M. I. T., Civil Engineering Systems Laboratory, May 1965

Weizenbaum, J., "Extrapolations", Conference on Computer Methods in the Analysis of Large-Scale Social Systems, Joint Center for Urban Studies, October 1964

Weizenbaum, J., "Computers and Behavioral Science", Technical Engineering News, January 1965

Weizenbaum, J., "Letter to the Editor on Implementations of SLIP", Communications of the ACM, May 1965

Wilde, D. U. and F. G. Ebaugh, Jr., "The Analysis of Red Blood Survival Curves", Vox Sanguineous, Vol. 10, No. 3, May 1965

Wilkes, M. V. "Slave Memories and Dynamic Storage Allocation", IEEE Transactions on Electronic Computers, Vol. EC-14, No. 2, April 1965

Witsenhausen, H., "Hybrid Solution of Initial Value Problems for Partial Differential Equations", AIAA Support for Manned Flight Conference, Dayton, Ohio, April, 1965 (AIAA Paper 65-262) (Also IFIPS Congress 1965, Vol. II, May, 1965)

Yngve, V. H., et al., "Group Report, COMIT", M.I.T. R.L.E. Quarterly Progress Report No. 76, January, 1965

Yntema, D. M., "The Software Problem", Lincoln Laboratory Group Report 1964-51, September, 1964

APPENDIX D
PROJECT MAC TECHNICAL REPORTS

REPORT NOS.	ASTIA NOS.	TITLE	AUTHOR(S)	DATE
MAC-TR-1	AD-604-730	Natural Language Input for a Computer Problem Solving Language	Bobrow, D. G.	6/64
MAC-TR-2	AD-608-499	SIR: A Computer Program for Semantic Information Retrieval	Raphael, B.	6/64
MAC-TR-3	AD-608-501	System Requirements for Multiple Access, Time-Shared Computers	Corbató, F. J.	5/64
MAC-TR-4	AD-604-678	Verbal and Graphical Language for the AED System: A Progress Report	Ross, D. T. Feldmann, C. G.	5/64
MAC-TR-6	AD-604-679	STRESS: A Problem-Oriented Language for Structural Engineering	Biggs, J. M. Logcher, R. D.	5/64
MAC-TR-7	AD-604-680	OPL-1: An Open Ended Programming System with CTSS.	Weizenbaum, J.	4/64
MAC-TR-8	AD-604-681	The OPS-1 Manual	Greenberger, M.	5/64
MAC-TR-11	AD-608-500	Program Structure in a Multi-Access Computer	Dennis, J.	5/64
MAC-TR-12	AD-609-296	The MAC System: A Progress Report	Fano, R. M.	6/64
MAC-TR-13	AD-609-288	A New Methodology for Computer Simulation	Greenberger, M.	10/64
MAC-TR-16	AD-612-702	CTSS Technical Notes	Saltzer, J. H.	3/65
MAC-TR-17	AD-462-158	Time-Sharing on a Multi-Console Computer	Samuel, A. L.	3/65
MAC-TR-18	AD-470-715	An Analysis of Time-Shared Computer Systems (thesis)	Scherr, A. L.	6/65
MAC-PR-1	AD-465-088	PROJECT MAC: Progress to July 1964 (participants)		

ARTIFICIAL INTELLIGENCE

BIOLOGY DEPARTMENT

CIVIL ENGINEERING DEPARTMENT

COMPUTER COMMUNICATION STRUCTURES

COMPUTER OPERATION

COMPUTER SYSTEM RESEARCH

ELECTRONIC SYSTEMS LABORATORY

LIBRARY RESEARCH

LINCOLN LABORATORY

NON-M.I.T. USERS

OPERATIONS RESEARCH CENTER

RESEARCH LABORATORY OF ELECTRONICS

SCHOOL OF ENGINEERING

SCHOOL OF HUMANITIES AND SOCIAL SCIENCES

SCHOOL OF SCIENCE

SLOAN SCHOOL OF MANAGEMENT

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Massachusetts Institute of Technology Project MAC	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED
	2b. GROUP None

3. REPORT TITLE

Project MAC
Progress Report II: July 1964 to July 1965

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

annual progress

5. AUTHOR(S) (Last name, first name, initial)

collection of material from many Project MAC participants;
no specific author or authors

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
1 July 1965	211	——

8a. CONTRACT OR GRANT NO. Office of Naval Research, Nonr-4102(01)	9a. ORIGINATOR'S REPORT NUMBER(S) MAC-PR-2
b. PROJECT NO. Nr-048-189	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency 3D-200 Pentagon Washington, D. C. 20301

13. ABSTRACT

The broad goal of Project MAC is experimental investigation of new ways in which on-line use of computers can aid people in their individual work; whether research, engineering design, management, or education.

This is the second annual Progress Report summarizing the research carried out under the sponsorship of Project MAC. The details of this research may be found in the publications listed at the end of the report.

14. KEY WORDS

Computers
Machine-aided cognition
Multiple-access computers
On-line computer systems
Real-time computer systems
Time-sharing
Time-shared computer systems

DD FORM 1 JAN 64 1473 (M.I.T.)

UNCLASSIFIED
Security Classification