AD NUMBER: AD0255504
LIMITATION CHANGES
TO: Approved for public release; distribution is unlimited.
FROM: Distribution authorized to U.S. Government agencies and their contractors; Administrative or operational use; 01 Mar 1961. Other requests shall be referred to US Air Force Institute of Technology, Wright-Patterson AFB, OH, 45433.
AUTHORITY
ST-A AFIT LTR, 19 AUG 1968

INSTITUTE OF TECHNOLOGY

AIR UNIVERSITY

UNITED STATES AIR FORCE

INSTITUTE OF TECHNOLOGY

SCHOOL OF ENGINEERING

THESIS

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

AUTOMATIC ANALOG COMPUTER PROGRAMING

THESIS

Presented to the Faculty of the School of Engineering of
the Institute of Technology (AU)
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

By

William J. Reisinger, B.S.
Major USAF

and

James C. Wine, B.S.
1st Lt. USAF

Graduate Guidance and Control
Department of Electrical Engineering

March 1961

Preface

The objective of this thesis is to establish the feasibility of using a digital computer to assist in programing an analog computer. The general principles and techniques are developed and discussed in Chapter II. The Automatic Program, which is one example of the implementation of these principles and which forms the basis for the digital computer program, is presented in Appendix A. The Automatic Program, a compilation of logic flow diagrams, was included as an appendix rather than in the body of the report for the convenience of the reader, since we believe that most readers are more interested in the basic ideas upon which the program is based rather than in the implementing details. However, the complete program may be useful for anyone interested in applying a similar approach, or a more general one, to a larger analog computer.

A great deal of the symbology involved in the program illustrated in Appendix A is entirely arbitrary and does not appear elsewhere in the report. Furthermore, many of the variables represented by the symbols are used in more than one sub-program. To avoid repetition of definitions, a fold-out containing a Glossary is included at the end of the report. An example of the application of this program to a problem is illustrated in Appendix B, to show typical results obtainable. Analog computer circuit diagrams representing manual and automatic programing results are sketched for comparison, and an actual sample of the digital computer output for the same problems is also included.

Appendix C represents a summary of knowledge gained during the initial attempt to formulate a more general program. We are hopeful that these results will prove useful to anyone interested in extending the techniques to more advanced systems.

Appendix D is a listing of the Fortran program written directly from the logic flow charts presented in Appendix A. This program is included for the benefit of the reader who may prefer the Fortran notation to that of the logic flow diagrams of Appendix A.

We are indebted to Mr. A. L. Robinson, Systems Dynamic Analysis Division, Directorate of Systems Engineering, Wright Air Development Division, Wright-Patterson Air Force Base, Ohio, who not only suggested the topic of this study, but started us in the right direction and offered helpful suggestions.

We are also deeply indebted to Captain Charles Whiting Richard, Jr., Assistant Professor of Mathematics, Institute of Technology, who provided enthusiastic support and interest and who shared our trials and tribulations during the stages of development of the topic. We also wish to express our sincere appreciation to Mr. Kenneth Bauman, who formulated the program in Fortran and exercised and tested it with sample problems.

Finally, we must acknowledge the extreme patience and understanding of our wives whose help in so many ways made the effort more enjoyable and fruitful.

William J. Reisinger
James C. Wine

Contents

List of Figures

Abstract

A digital program is developed to assist the engineer in preparing a problem for solution on an analog computer. The program is based on a problem input which consists of a first-order system of homogeneous linear equations with constant coefficients in matrix form, and the maximum values of the problem variables. The program implements procedures to examine the system coefficient matrix in order to check the input for compatibility with program limitations; amplitude- and time-scale the problem to fit the computer equipment used; assign additional amplifiers to eliminate integrator output loading effects and form the required negative integrator outputs; calculate potentiometer gain settings; and specify the assignment of integrators, amplifiers, and potentiometers necessary for the solution of the problem. The output of the digital computer is in the form of printed data which includes explicit instructions for wiring integrators, amplifiers, and potentiometers together; integrator input gains; potentiometer settings; time- and amplitude-scale factors; and data to correlate the information with the original problem.

AUTOMATIC ANALOG COMPUTER PROGRAMING

I. Introduction

The object of this report is to describe the development of a technique for using the digital computer to assist the engineer to program problems for solution on an analog computer. The technique should prove the feasibility of using the digital computer for this application and provide the basis for more advanced programing systems in this field.

The Need for Automatic Analog Computer Programing

In order to approach this object in a systematic manner, it is necessary to consider the characteristics of both analog and digital computers. The analog computer is particularly well-suited to the solution of differential equations. In most analog computers, the problem variables are represented by voltages. The integrations necessary to solve differential equations are, therefore, easily performed by amplifiers which sum, over an interval of time, the voltage present at their inputs. The differential equations representing the problem to be solved by the computer are programed on the computer by a wired patchboard. The connections on the patchboard interconnect the units of the computer in such a manner that the problem solution is obtained. Since each operation indicated by the equations is represented by a physical unit of the computer, the solution of more complicated systems of equations requires more equipment and connections on the patchboard than do simple systems.

As the complexity of problems which may be represented by differential equations has increased, larger analog computers have been constructed. The net result is that considerable engineering effort is required to program the large-scale analog computer problems. Much of this programing effort consists of operations with algebraic quantities and repetitive logic. The digital computer is ideally suited to the handling of these types of operations. It will perform arithmetic and comparative operations on numerical quantities at the rate of several thousands per second. While digital computers are not as generally available as are analog machines, most of the computing centers operating the large analog computers also have digital computing facilities. It therefore appeared desirable to investigate the possibility of using the digital computer to perform the analog computer programing. As a literature search revealed no reference to such a technique, it was decided to develop the system described in this report.

Scope of the Automatic Program

While it would be desirable to have a digital computer system which would produce an analog computer program for any problem within the capability of the analog machine, such an ambitious project is beyond the scope of this investigation. The present study will, therefore, be limited to the programing of systems of ordinary homogeneous linear differential equations with constant coefficients. It is anticipated, however, that

The analog computer for which the program was developed, Reeves Electronic Analog Computer Model 101, is divided into two bays, each containing identical components in symmetrical arrangement.

Units Available:

Unit	Quantity*	Inputs	Outputs
Integrators	14	2 gain tens 2 gain fours 3 gain ones	five
Summing Amplifiers	14	2 gain tens 2 gain fours 3 gain ones	four
Inverting Amplifiers	12	3 gain ones	four
Potentiometers	46	one	one
Interconnections	44	---	---

* Total of both bays

Characteristics:

Usable Variable Range: 0.01 to 100 (volts)

Usable Potentiometer Settings: 0.05 to 1.00

Usable Frequency Range: 0.12 to 18.8 rad/sec

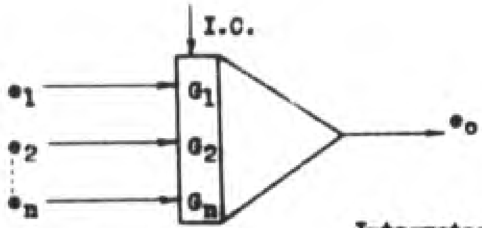
Figure 1

Analog Computer Characteristics

many of the procedures used with these equations may be extended, by suitable modifications, to include other types of differential equations.

During the earlier phases of this investigation, the general problem was arbitrarily defined to encompass systems of linear equations up to and including fourteenth order. This limit was chosen on the basis of using two units of the Reeves Electronic Analog Computer available at the Institute of Technology. For any given computer there are practical limitations on (1) the magnitudes of the output voltages and currents of any amplifier; (2) the response rate of the components; (3) the number and magnitude of the gains available at the inputs of the amplifiers and integrators; and (4) the number of components available. Figure 1, facing, summarizes the important characteristics of the Reeves Model 101 computer used. It was felt that an automatic system capable of handling a problem of this magnitude could easily be extended to larger systems if the additional analog equipment were available.

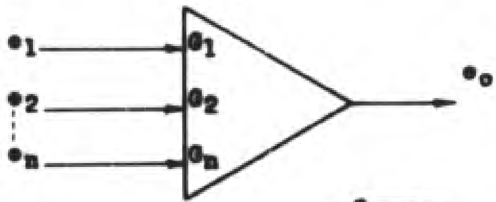
Inasmuch as a maximum of seven inputs were available on the integrators of the analog computer, it was necessary to program for as many as 14 inputs by combining or summing on amplifiers connected to the inputs of the integrators. It was soon apparent that techniques to handle the many combinations of units which were possible, along with the problems imposed by the overloading effects on the summing amplifiers, would result in considerable complexity in the final automatic program. On the basis of recommendations made by the project sponsor, Mr. A. L. Robinson, it was decided that a more practical objective



Integrator

$$e_o = -\int_0^t (G_1e_1+G_2e_2+ \dots +G_ne_n)dt + I.C.$$

where G_i = integrator gain

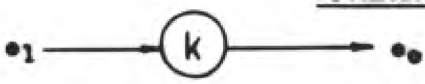


Summer

$$e_o = -(G_1e_1+G_2e_2+ \dots +G_ne_n)$$

where G_i = summer gain

Potentiometer



$$e_o = ke_1 \text{ ---- } 0 \leq k \leq 1$$

Figure 2

Analog Computer Components

for this initial effort would be to include linear systems up to and including fourteenth order which would result in requirements for no more than seven inputs to any integrator and no more than seven outputs from any integrator. A program based on these limitations is applicable to nearly all of the problems of this type normally encountered in practice, and should therefore not seriously jeopardize the generality of the results. However, a summary of the ramifications and requirements of the more general problem is included in Appendix C for those readers who may be interested in extending and expanding the technique.

Within the framework outlined above, it has been assumed that (1) the given problem can be expressed as a system of first-order equations in matrix form, and that the problem is so formulated by the operator before it is presented to the digital computer; and (2) the maximum values of the problem variables are known and are presented to the digital computer as input information.

Problem Analysis

In general, any system of linear differential equations with constant coefficients may be solved on an analog computer by interconnecting three types of computer components; namely, summing integrators, summing inverters, and potentiometers. The independent variable is represented by time and the dependent variables are expressed in terms of voltages. A common diagrammatic representation of the three analog computer components together with their input and output voltages are shown in Figure 2, facing. The inputs and outputs of these components are wired internally to a patchboard, and the components may be interconnected by external wires which are plugged into the patchboard.

$$
\begin{aligned}
-\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\
-\dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\
&\vdots \\
-\dot{x}_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n
\end{aligned}
$$

where $\dot{x}_i \equiv dx_i/dt$

in matrix form

$$
-\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \cdots a_{1n} \\ a_{21} & a_{22} \cdots a_{2n} \\ \vdots & \vdots \\ a_{n1} & a_{n2} \cdots a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}
$$

or $-\dot{X} = AX$

Figure 3

First Order System and Matrix Equivalent

Since a system of linear homogeneous equations to be programed may represent a wide variety of physical phenomena, it may consist of any number of equations of any order. It is possible, however, to transform such equations into a first-order system which may then be written in matrix form. An example of a first-order system of n homogeneous equations together with its matrix equivalent is illustrated in Figure 3, facing. Thus, it is necessary only that the program be capable of handling a system of first-order linear equations in matrix form. To develop such a program it will be necessary to solve a number of problems.

Scaling. The first task in preparing these first-order equations for solution on the analog computer reduces to one of developing automatic procedures to (1) amplitude-scale the equations -- perform consistent changes of variable to limit the maximum amplitudes of the problem variables to those possible on the analog computer; and (2) time-scale the equations -- change the problem rates to those within the capabilities of the computer.

Integrator Loading Effects. The possibility of introducing degenerate effects as a result of loading integrators with too many potentiometers, as well as the requirement for forming negative inputs to the integrators implies a need for adding amplifiers to the integrator outputs. It is therefore necessary to develop procedures to (1) determine the need for the amplifiers at the integrator outputs, and (2) determine the optimum configuration of the additional equipment which will most economically satisfy this need.

Equipment Selection. Procedures must also be developed to solve the logical problem of selection of individual analog computer units and the specification of patchboard connections. These techniques must result in solutions which are physically possible with the analog equipment being used and should result in its economical use.

Digital Computer Output. The remaining problem is the development of the digital computer output program. This program must present in a clear, readily understandable form, the results of the programing decisions made by the digital computer. The output must contain patchboard wiring instructions, scale factors, potentiometer settings, and information relating the potentiometers to the original coefficient matrix elements.

Plan of the Report

Each of the issues will be investigated and procedures to resolve them developed, discussed, and exemplified. The result will be a series of systematic techniques which, when embodied in a digital computer program will (1) check a given problem for compatibility with the limitations of the analog computer; (2) amplitude- and time-scale the problem (when necessary); and (3) provide printed information which will specify the assignment of analog equipment required to solve the problem, along with other pertinent data needed by the analog computer operator. A series of detailed logic flow charts (programs) are developed which indicate the implementation of the techniques on the digital computer. Instructions for the use of the digital system will be presented along with an example problem which indicates the results of the techniques. Finally, the efficiency of the automatic program will be compared to that of the manual procedure, and recommendations will be made for extending the basic ideas to the development of a more general program.

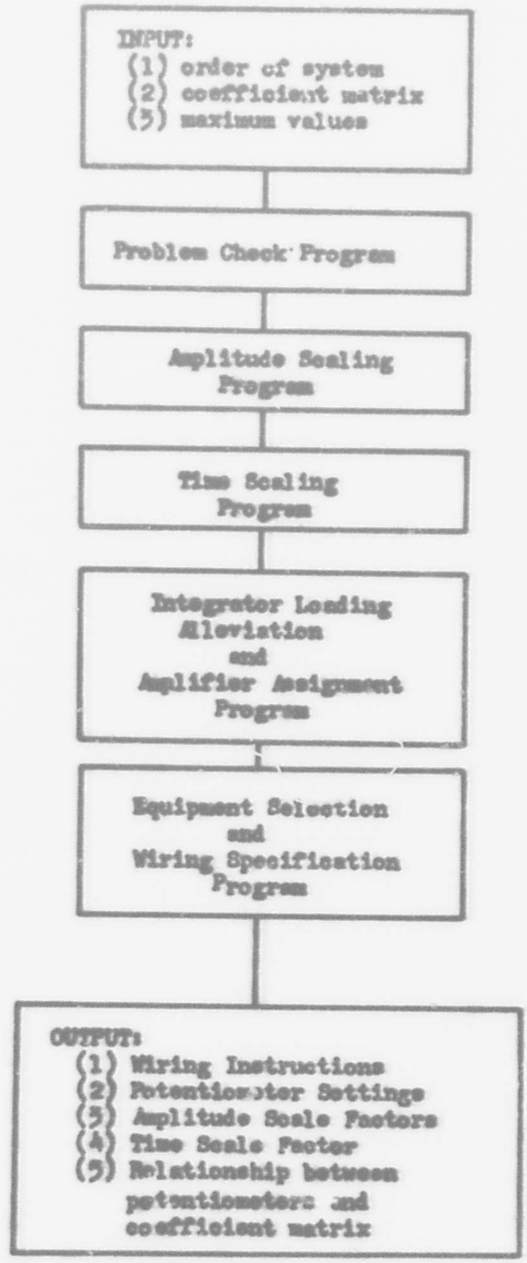


Figure 4
The Automatic Program

II. Development of the Automatic Program

As was noted earlier, the digital computer is ideally suited to the solution of the problems encountered in preparing a system of equations for the analog computer. However, the efficient use of the digital computer requires that all problems to be programed be presented in a standard form. Fortunately, all of the equations within the scope of this investigation may be reduced to a system of first-order equations. It becomes convenient, therefore, to specify that the input to the digital computer should be the first-order reduction, in matrix form, of the set of problem equations. The matrix form was chosen primarily because of the ease with which these equations may be manipulated by the digital computer.

The standard form selected for the input equations is the matrix equation,

$$-\dot{X} = AX \qquad (2\text{-}1)$$

Where, for an nth order system, $\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n \end{bmatrix} \qquad X = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$

Consider the fourth-order linear homogeneous equation with constant coefficients

$$a\ddddot{x} + b\dddot{x} + c\ddot{x} + d\dot{x} + e x = 0$$

Letting $x_1 = x; \quad x_2 = -\dot{x}_1; \quad x_3 = -\dot{x}_2; \quad x_4 = -\dot{x}_3$

and substituting,

$$-a\dot{x}_4 - bx_4 + cx_3 - dx_2 + ex_1 = 0$$

Solving for $-\dot{x}_4$,

$$-\dot{x}_4 = -ex_1/a + dx_2/a - cx_3/a + bx_4/a$$

Summarizing

$$-\dot{x}_1 = x_2$$

$$-\dot{x}_2 = x_3$$

$$-\dot{x}_3 = x_4$$

$$-\dot{x}_4 = -ex_1/a + dx_2/a - cx_3/a + bx_4/a$$

In matrix form

$$-\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{e}{a} & \frac{d}{a} & -\frac{c}{a} & \frac{b}{a} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Figure 5

Conversion of Equation to Matrix Form

and

$$A = \begin{bmatrix} a_{11} & a_{12} \cdots & a_{1n} \\ a_{21} & a_{22} \cdots & a_{2n} \\ \cdot & \cdot & \cdots \cdot \\ \cdot & \cdot & \cdots \cdot \\ a_{n1} & a_{n2} \cdots & a_{nn} \end{bmatrix}$$

An example of the procedure to reduce a typical problem to this standard form is presented in Figure 5, facing, for the reader who may not be familiar with this type of conversion. It may be noted that, with the exception of the maximum values of the dependent variables, all of the information needed to develop the analog circuit for any given problem is contained in the coefficient matrix (A). That is:

(1) The non-zero elements of each column of the coefficient matrix represent the outputs of the corresponding integrators.

(2) The non-zero elements of each row of the coefficient matrix represent the inputs to the corresponding integrators.

(3) The magnitude of each element of the coefficient matrix represents the gain required in the path from an integrator output (indicated by the column containing the element) to an integrator input (indicated by the row containing the element). It may be noted that this gain, in the analog computer, is composed of the integrator gain and a potentiometer gain.

Problem Summary

Given: (1) Problem System of first-order linear homogeneous equations in matrix form ($\dot{X} = AX$).

(2) Maximum values of the dependent variables ($x_{i\ max}$; $i = 1 \rightarrow n$)

To determine:

(1) Amplitude scale factor (S_i)

(2) Time scale factor (λ)

(3) Integrator - amplifier combinations which will provide the necessary positive and negative outputs specified by the coefficient matrix (A).

(4) Potentiometer settings $P_{ij} = (a_{ij} S_i / S_j)/G_I$

(5) Proper combination of integrators, amplifiers, potentiometers, and associated wiring to effect correct analog circuit for problem solution.

(6) Digital computer output in printed form specifying data and/or instructions contained in (1) through (5).

Problem Restrictions:

Input: System of linear homogeneous differential equations up to and including fourteenth order provided that:

(1) not over 7 inputs are required to any integrator,

(2) not over 7 outputs are required from any integrator-amplifier combination, and

(3) not over 46 total inputs are required to all integrators, i.e. not over 46 potentiometers are required.

Equipment: The following equipment restrictions are incorporated in the program, but introduce no restrictions on the input problem beyond those indicated above.

(1) Fourteen integrators are available.

(2) Integrator inputs are restricted to the set (10,10,4,4,1,1,1).

(3) Integrator outputs are limited to feeding not more than four potentiometers and one recorder.

(4) Fourteen summing amplifiers and twelve inverting amplifiers are available.

(5) Amplifiers required to form additional outputs and/or sign changes are assigned to integrator outputs at gain one.

(6) Amplifier outputs are limited to feeding not more than four potentiometers.

(7) Potentiometers are limited, for convenience, to one output.

(8) Fourty-four interconnections are available.

Figure 6
Problem Summary and Restrictions

(4) A negative sign associated with an element of the coefficient matrix indicates that an inverting amplifier is required in the path from integrator output to integrator input.

Consequently, the only inputs required to the digital program are the order of the system being considered, the coefficient matrix, and the maximum values of the dependent variables. The problem, then, in general, is to develop procedures by which the digital computer can automatically check the input for operator mistakes, calculate the amplitude- and time-scale factors, determine the inverter configuration which will reduce the number of components required to solve the problem, assign the appropriate equipment, and print out the information which the analog computer operator needs to wire and investigate the problem. In the development to follow, each of these requirements will be analyzed and discussed from a general viewpoint. Related problem restrictions, if any, will then be introduced and explained, and finally a solution will be presented. The net result will be one solution to the general problem, subject to the specified restrictions. The problem and the limiting restrictions are summarized in Figure 6, facing.

<u>Problem Check</u>

Even though it is assumed that the operator will not present problems which exceed the capabilities of the automatic program, it was deemed advisable to incorporate tests which would identify problems which are unsolvable by the system. Therefore, a simple procedure was devised to provide a means of rejecting a given problem in cases where: (1) more than seven inputs to any integrator are indicated and (2) more than 46 potentiometers are required to form the inputs specified by the coefficient matrix. (A provision is made in the Amplifier Assignment Program to reject a problem which violates the restriction on the number of outputs.)

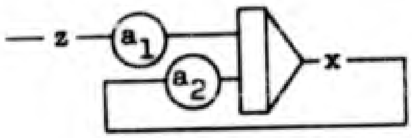
The Problem Check Program, Appendix A, (Figure A-1), implements the first of these checks by specifying a count of each row of the matrix to determine whether more than seven non-zero elements exist. If any row of the coefficient matrix contains more than seven non-zero elements, the program is stopped and a print-out statement specifies the reason for rejection. The second check is implemented by summing the non-zero elements in all rows. If the total number is greater than 46, the program is stopped and the reason printed out.

Since this initial program requires that the entire matrix be examined, it was convenient to incorporate instructions for recording the absolute values of the maximum and minimum (non-zero) elements of the matrix. These values are required later in the time-scaling routine.

Amplitude Scaling

In general, the technique of amplitude scaling consists of a uniform change of problem variables so as to limit their maximum amplitudes to values which are attainable on the analog computer. The effect of amplitude scaling upon a system of equations will be reviewed, and the technique will be extended to a general system of linear equations in matrix form. Procedures will be developed to determine the optimum amplitude scale factors and to apply them in the formation of the scaled-matrix system.

Amplitude Scaling Applied to Matrix Equations. As a brief review of amplitude scaling, consider the following portion of a computer circuit:



If it is desired to scale the magnitude of x to make its maximum value ten times larger, we write 10x. Then,

$$x = \int +\dot{x}\, dt \tag{2-2}$$

$$10x = \int +10\dot{x}\, dt \tag{2-3}$$

But
$$-\dot{x} = a_2 x + a_1 Z \tag{2-4}$$

and
$$-10\dot{x} = a_2(10x) + (10a_1)Z \tag{2-5}$$

Therefore, in this simple example, it may be observed that when x is scaled by ten, $\dot{x}$ is also scaled by ten. Further, a_1, the coefficient of Z, must be scaled by ten to preserve the value of the equation.

Extending these principles to a system of equations of the form,

$$\begin{aligned} -\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ -\dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \\ -\dot{x}_3 &= a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \end{aligned} \tag{2-6}$$

and scaling x_1 by S_1, x_2 by S_2, and x_3 by S_3; there results,

$$\begin{aligned} -S_1\dot{x}_1 &= a_{11}(S_1x_1) + S_1a_{12}x_2 + S_1a_{13}x_3 \\ -S_2\dot{x}_2 &= S_2a_{21}x_1 + a_{22}(S_2x_2) + S_2a_{23}x_3 \\ -S_3\dot{x}_3 &= S_3a_{31}x_1 + S_3a_{32}x_2 + a_{33}(S_3x_3) \end{aligned} \tag{2-7}$$

The scaled equations may then be written in matrix form.

$$-\begin{bmatrix} S_1\dot{x}_1 \\ S_2\dot{x}_2 \\ S_3\dot{x}_3 \end{bmatrix} = \begin{bmatrix} a_{11} & (S_1/S_2)a_{12} & (S_1/S_3)a_{13} \\ (S_2/S_1)a_{21} & a_{22} & (S_2/S_3)a_{23} \\ (S_3/S_1)a_{31} & (S_3/S_2)a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} S_1x_1 \\ S_2x_2 \\ S_3x_3 \end{bmatrix} \tag{2-8}$$

As is readily seen, the general element of the amplitude-scaled matrix may be expressed as

$$E'_{ij} = \frac{S_i}{S_j} a_{ij}. \tag{2-9}$$

By the rules of matrix multiplication it can be shown that the amplitude-scaled system of equations may be written in the following form:

$$-S\dot{X} = SAS^{-1} \cdot SX \tag{2-10}$$

where $$S = \begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{bmatrix} \qquad (2\text{-}11)$$

Determination of Amplitude Scale Factor. This amplitude scaling equation in matrix form can be applied only after the values of the elements of the S matrix are determined. It will be recalled that one of the items of information available concerning the problem was the maximum value of each variable. It is apparent that with this knowledge and the maximum useable value of machine variable, the magnitude of each element of the S matrix may be calculated. Thus, using the convention that one unit of problem variable equal one machine volt

$$S\,X_{max} = (\text{maximum machine voltage}) = 100 \qquad (2\text{-}12)$$

or $$S_i = 100/|x_i|\,max \qquad (2\text{-}13)$$

As is apparent, the direct application of the preceding equation will, in general, yield non-integer values for the scale factors which will be rather difficult to use because of their effect upon the scaling of the output recordings. The output scaling problem may be alleviated, however, at only a small sacrifice in the optimum machine scaling, by restricting the possible values of the elements (S_i) to the form

$$S_i = B.10^c \qquad (2\text{-}14)$$

where B = integer values from 1 to 9

c = positive or negative integers, including zero.

The amplitude scale factor will then be the largest value satisfying the relation

$$S_i = B\cdot 10^c = 100/|x_i|_{max} \qquad (2\text{-}15)$$

These scale factors may be used to form the S matrix of Equation (2-11) which is required in the amplitude-scaled coefficient matrix of Equation (2-8).

The general technique explained in the preceding paragraphs has been implemented in the form of a logic flow chart which comprises the Amplitude Scaling Program, Appendix A (Figure A-2). It includes a systematic routine which specifies the operations which are necessary to determine the optimum amplitude scale factors for a given coefficient matrix, and it specifies the program required to form the amplitude-scaled matrix equation.

Time Scaling

For an analog computer simulation to be accurate, the problem frequency range must be compatible with the machine solution rate. Therefore, time scaling may be necessary in addition to amplitude scaling. This is done by changing the independent variable (time) in such a manner that machine-variable rates remain proportional to the original problem-variable rates.

Application of Time Scaling to Matrix Equations. Suppose that a problem is to be speeded λ times. Then,

$$t = \lambda \tau \tag{2-16}$$

where t = real time (independent variable)

τ = machine time

and λ will be a quantity greater than one. If λ is less than unity the problem will be slowed $1/\lambda$ times. Therefore,

$$dx/dt = 1/\lambda \; dx/d\tau \tag{2-17}$$

$$\dot{x} = (1/\lambda)x' \tag{2-18}$$

where $x' = dx/d\tau$

Then, as before, considering Equation (2-6) upon which it is desired to perform a time scale change of λ,

$$\dot{x}_1 = (1/\lambda)x_1'$$

$$\dot{x}_2 = (1/\lambda)x_2' \qquad (2\text{-}19)$$

$$\dot{x}_3 = (1/\lambda)x_3'$$

Thus in matrix form, the system of equations becomes

$$-\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = -\begin{bmatrix}(1/\lambda)x_1'\\ (1/\lambda)x_2'\\ (1/\lambda)x_3'\end{bmatrix} = -(1/\lambda)\begin{bmatrix}x_1'\\ x_2'\\ x_3'\end{bmatrix} = \begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} \qquad (2\text{-}20)$$

or

$$-\dot{X} = -(1/\lambda)X' = AX \qquad (2\text{-}21)$$

Multiplying the above equations by λ gives

$$-\lambda\dot{X} = -X' = \lambda AX \qquad (2\text{-}22)$$

Applying the time scale factor to the amplitude-scaled equation, Equation (2-10),

$$-SX' = -S\lambda\dot{X} = \lambda SAS^{-1} \cdot SX \qquad (2\text{-}23)$$

Selection of the Time Scale Factor. If the roots of the characteristic equation of the input system of equations are known, the frequencies present in the problem can be determined. The appropriate time-scale factor may then be selected on the basis of the maximum and minimum natural frequencies of the problem and the useful range of machine frequencies. If, on the other hand, as is usually the case, the roots of the characteristic equation are not known, considerable labor is involved in their determination. With the system of equations being considered, it is possible to extract the roots of the characteristic equation by expanding its determinant and factoring the resulting equation.

Fortunately, however, for the purpose of time-scaling, the maximum frequency present in the solution of the equations is of primary interest. The maximum frequency is used as the basis for making initial time-scale changes and is, therefore, the controlling frequency in most problems. Only when the maximum frequency is originally within the useful range of computer frequencies, but not at the maximum usable frequency, is the minimum problem frequency of interest. In this instance the decision to time-scale, or not, is based upon the minimum frequency.

For a coefficient matrix such as the one under consideration, it may be shown that the maximum frequency present is less than (or equal to) the maximum sum of the elements of each row (Ref 1).

That is,
$$w_{n\,max} = \max \sum_{j=1}^{N} |a_{kj}| \qquad (2\text{-}24)$$

It may be observed, moreover, that in the majority of practical systems which may be expressed in matrix form, there will be relatively few non-zero elements in most rows of the coefficient matrix. It is therefore reasonable to assume that the maximum frequency is roughly indicated by the maximum element in the matrix. Or, in mathematical form,

$$w_{n\,max} \approx |a_{ij}|\,max \qquad (2\text{-}24)$$

Since the maximum gain present at the input of any integrator is ten, and since the magnitude of the output of each of the integrators has been set at, or near, 100 volts, it follows that the gain possible in any simple loop of the computer circuit--hence, the maximum value of any coefficient in the scaled matrix -- is restricted to a value of ten. This restriction, coupled with the approximation indicated by Equation (2-24), implies that the maximum frequency present in the machine problem, which has been scaled so that the gains required are within the range obtainable on the computer, will be about 10 radians per second. Since the analog computer used in this study has a range of problem frequencies from 0.12 to 18.8 radians per second, even a 50% variation in the approximation represented by Equation (2-24) will not invalidate the results.

The choice of time-scale factor will depend ultimately upon the availability of specific values of gain to form the required inputs to each integrator. In general, it will be of the form,

$$\lambda |a_{ij}| = G_I \qquad (2\text{-}25)$$

where a_{ij} is a matrix element which must be scaled to make it possible to form all of the inputs to a given integrator with the gains available. It is pertinent to note that it may be necessary to apply Equation (2-25) a number of times in order to find a final time scale factor which will result simultaneously in compatibility with the gains available and with the inputs to each integrator.

The procedure for arriving at the final time scale factor (implemented by the Time Scale Program of Appendix A) is explained in the following paragraphs.

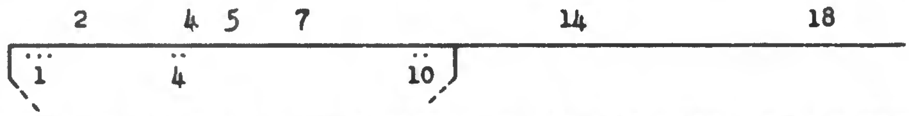
1. A need for a preliminary time scale change is determined by consideration of the maximum and minimum values of the coefficient matrix. In the analog computer used for this study, if the magnitude of the maximum element is less than 10 and if the magnitude of the minimum (non-zero) element is not less than 0.10, no preliminary time scaling is required ($\lambda_1 = 1$).

2. If the above conditions are not satisfied, a trial time scale factor, λ_1, is calculated by applying Equation (2-25), using a gain of ten and the magnitude of the largest matrix element.

3. In either case, each row of the matrix is checked to insure that no more than two inputs at gain ten are required on any integrator. This check is made by inquiring whether the third largest element in the row being considered is greater than $4/\lambda$, where λ in all cases is the last calculated value. If not, no change is required as a result of this test. If the third largest element of any row is greater than $4/\lambda$ (if the test fails), more than two inputs at gain ten are required, and the calculation of another trial time scale factor is necessary. This factor is obtained by again using Equation (2-25) with a gain of 4 and the element of the matrix failing the above test. The program is then continued using this new time-scale factor.

4. After insuring that no more than two gains of ten are required on any integrator, a similar check is made to insure that no more than four gains of four and ten are required on any integrator. This check is made by inquiring whether the fifth largest element in any row is greater than $1/\lambda$. If the element is greater than $1/\lambda$, a new time-scale factor is calculated by again applying

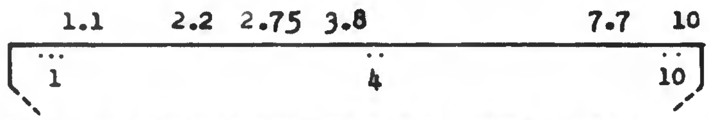
Given the hypothetical matrix row shown in relation to the gain-range available on the associated integrator,



Since at least one element of the row is greater than the largest gain available, the row must be scaled to make the largest element equal or less than 10. Applying Equation 2-25 (Step 2),

$$\lambda_1 = a_I / |a_{ij}|\max = 10/18 = 0.55$$

The result of applying the scale factor is equivalent to the new range of variables shown.



To insure that no more than two inputs at gain 10 will be required Equation 2-25 (Step 3) is applied using the third largest element and a gain of four. Thus we require that,

$$4/\lambda_1 \geq 7$$

$$7.27 > 7$$

The result indicates that the requirement has been satisfied. A similar check is then made to insure that no more than four gains four and ten are needed (note that the diagram above reveals that six would be required). Thus we require that (Step 4)

$$1/\lambda_1 \geq 4$$

$$1.81 \not\geq 4$$

Since the test failed, a new time-scale factor must be calculated. Applying Equation 2-25 again,

$$\lambda = 1/4 = 0.25 \text{ (final)}$$

Applying this scale factor to the row shown above, there results,

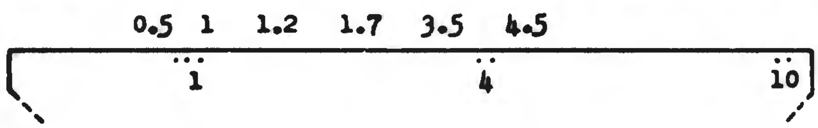


Figure 7

Time Scaling

Equation (2-25), using a gain of one and the element of the matrix upon which the test failed.

5. The last time-scale factor calculated in proceeding through the above tests is the final time-scale factor. This procedure is summarized with an abbreviated example in Figure 7, facing, and implemented by the Time Scaling Program of Appendix A (Figure A-4, Part II). It is apparent, however, that the time-scale factor calculated by the procedure described may be a non-integer. If this were the case, it would be inconvenient for the analog computer operator to use in the final interpretation of the problem solution. A time-scale factor is needed which will facilitate the conversion of the solution from machine time to real time associated with the physical problem. For this reason it was considered desirable to round-off the final time-scale factor by restricting it to the form

$$\lambda = b \cdot 10^{c} \leq T_1(\text{final}) \qquad (2\text{-}26)$$

where

b=a positive integer: 1,2,3,....,9.

c=positive or negative integer(s), including zero.

6. The final step in time-scaling the problem is the formation of the time-scaled matrix by multiplying each element in the amplitude-scaled matrix by the time scale factor, λ .

It may be noted that some of the steps in the time-scaling program discussed above imply the need to pick out any element of a row of the coefficient matrix on the basis of its relative magnitude with respect to the other elements of that row, i.e., that the matrix rows have been ordered. This was accomplished, as is indicated in the Time Scale

Program of Appendix A (Figure A-3, Part I), by generating a matrix in which the elements of each row identify the elements of the corresponding row of the amplitude scaled matrix in descending order of magnitude. The elements of each row of the new matrix (G) are the column subscript (J) values of the elements of each row of the E' matrix, arranged from left to right so that the J value of the largest element appears first, the next largest, second, etc. In other words, the new matrix is a matrix composed of the columnar subscripts of the E' matrix, arranged so as to specify the elements of each row of the E' matrix in descending order of magnitude. For example, to find the magnitude of the second largest element in the fourth row of the coefficient matrix, the computer would find the 'value' of the second element of the fourth row of the matrix of subscripts. The 'value' of this element would be the column subscript of the element of the coefficient matrix whose magnitude was second from the largest in that row. Such an elaborate means of ordering the matrix was necessary since, as was indicated earlier, not only the magnitudes of the elements of the coefficient matrix, but their position as well, are important in the solution of the problem.

The result of the time scale program described above is a system of equations which is compatible with the analog computer insofar as the maximum values of the variables and the problem frequencies are concerned. In addition, the inputs to the various integrators are compatible, in magnitude, with the gains available.

Integrator Loading Alleviation and Amplifier Assignment

Having devised procedures for scaling the system variables to insure that the maximum amplitudes and frequencies do not exceed machine limitations, it becomes necessary to consider the integrators and amplifiers required for the solution of the problem.

<u>Requirement for Inverting Amplifiers</u>. If ideal analog computer components are assumed, the only requirement for inverting amplifiers is to provide the sign changes indicated by the coefficient matrix. In this instance, the necessary amplifiers may be placed at either the integrator inputs or the integrator outputs. The most advantageous location for a particular amplifier will depend upon the problem conditions as specified by the coefficient matrix. It may be noted that, as described in Appendix C, it is possible in some cases to reduce the number of inversions required by suitable variable changes. The information necessary to determine whether a lesser number of amplifiers will be required for a given problem if they are placed either at the integrator inputs or outputs may be obtained from the coefficient matrix.

After suitable changes of variable, as indicated in Appendix C, let C_- be the minimum number of columns of the matrix containing negative elements,

R_- be the minimum number of rows of the matrix containing negative elements,

then if $R_- < C_-$, a lesser number of amplifiers will be required if they are assigned to the inputs of the integrators associated with the rows containing negative elements, and conversely.

The technique discussed above will permit the assignment of a minimum number of inverting amplifiers to provide a solution to a problem (assuming ideal analog computer elements). Unfortunately, however, any practical analog computer will introduce limitations and restrictions which will require the modification of the inverting amplifier assignment criteria.

First, any physical amplifier will have an upper limit of usable output voltage. The possibility therefore exists that this limit will be exceeded when several variables are summed on an amplifier. This is the case when amplifiers are assigned to the integrator inputs. The fact that the integrated sum of the inputs has been scaled provides no guarantee that the sum of any arbitrary set of these inputs will not exceed the maximum limit of permissible voltage. As an illustration, consider an amplifier connected to an integrator input to form the inputs specified by the k-th row of the scaled coefficient matrix. Let,

e_{kj}=general element of the k-th row of the matrix

P_{kj}=Potentiometer setting for general element

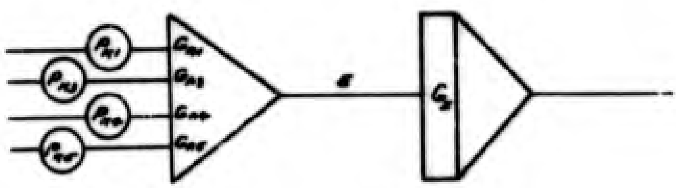
G_A =input gains of amplifier A

G_I = input gain of integrator

$X_k(max)S_k = X_{km}S_k$=maximum value of general variable
100 volts

E =100volts=maximum permissible output of amplifier

and assume that e_{k1}, e_{k3}, e_{k4}, and e_{k5} are negative.



Then for the negative elements,

$$e_{kj}=P_{kj}G_AG_I \qquad (2\text{-}27)$$

or

$$P_{kj}=ekj/(G_AG_I) \qquad (2\text{-}28)$$

Therefore,

$$P_{k1}G_{A1}S_kX_{1m}+P_{k3}G_{A2}S_kX_{3m}+P_{k4}G_{A3}S_kX_{4m}$$
$$+P_{k5}G_{A4}S_kX_{5m} \leq E_m=100 \qquad (2\text{-}29)$$

or

$$(e_{k1}/G_I)X_{1m}+(e_{k3}/G_I)X_{3m}+(A_{k4}/G_I)X_{4m}$$
$$+(A_{k5}/G_I)X_{5m} \leq 100 \qquad (2\text{-}30)$$

But

$$X_{1m}\approx X_{3m}\approx X_{4m}\approx X_{5m}\approx 100 \qquad (2\text{-}31)$$

Thus:

$$e_{k1}100+e_{k3}100+e_{k4}100+e_{k5}100 \leq 100G_I \qquad (2\text{-}32)$$

or

$$e_{k1}+e_{k3}+e_{k4}+e_{k5} \leq G_I \qquad (2\text{-}33)$$

Therefore

$$\sum(e_{kj}<0) \leq G_I \qquad (2\text{-}34)$$

The result above reveals that to avoid overloading the amplifier assigned to the input of an integrator, the summation of the coefficient matrix elements associated with the inputs to that amplifier must be less than the integrator gain fed by that amplifier. Further, Equation (2-28) shows that the potentiometer setting is a direct function of the integrator gain. Therefore, to assign amplifiers to the integrator inputs, it would be necessary to satisfy the overload restriction (Equation 2-34) without generating a need for excessively low potentiometer settings.

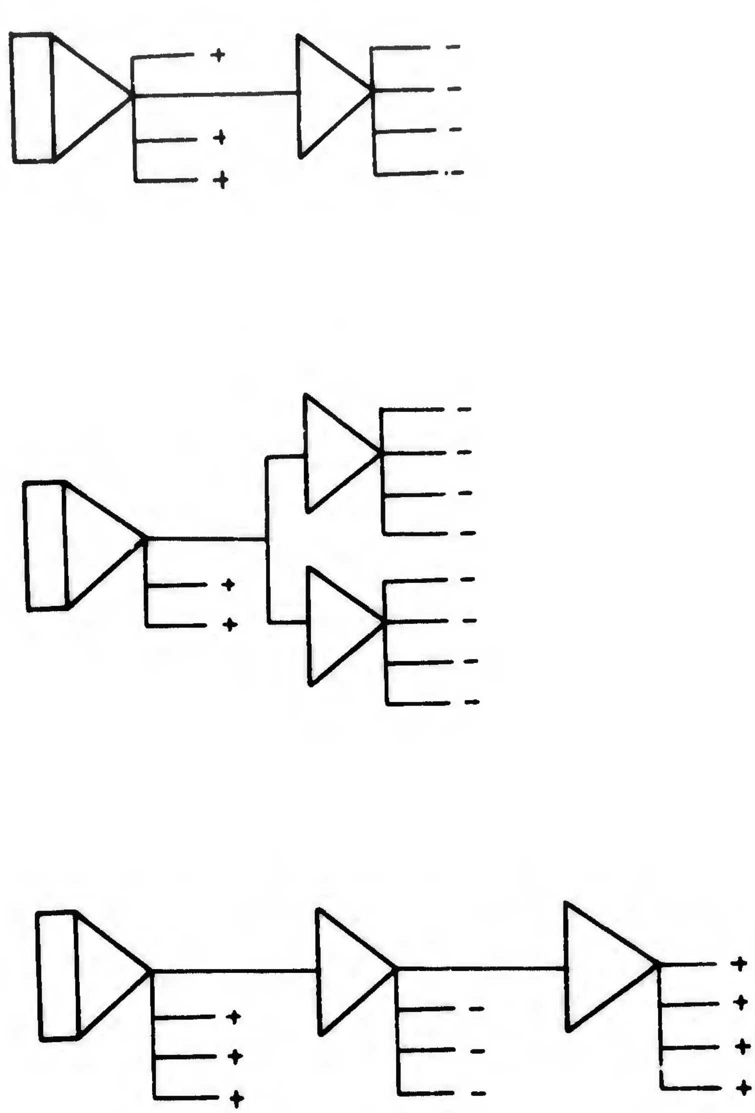


Figure 8

Integrator - Amplifier Combinations

A second characteristic of the analog computer equipment which further complicates any procedure for determining the minimum number of inverting amplifiers is that of integrator output loading. For instance, the integrators and amplifiers of the Reac Model 101 were found to be capable of supplying no more than four potentiometers. Thus, any integrator of this computer which was required to supply more than four outputs would require the addition of an inverting amplifier.

It follows, therefore, that for the analog computer used, any column of the coefficient matrix containing more than four non-zero elements implies a requirement for the addition of an amplifier or amplifiers. Since these amplifiers also provide sign changes, their use to alleviate integrator overloading prevents the direct application of the techniques, described above, to determine the minimum number of inverters required.

Inverting Amplifier Assignment. It was decided, for this initial effort, to write a program which would assign amplifiers only to the integrator outputs. Such a program will, in some cases, require the use of more amplifiers than would a more sophisticated technique, but it will in no case exhaust the total number of amplifiers (inverters and summers) available. It follows, therefore, that any column which contains more than four non-zero elements and/or any negative elements implies a requirement for the addition of an amplifier (or amplifiers) to the output of the associated integrator.

The number and specific configuration of the added amplifiers will therefore depend upon the total number of non-zero elements and upon the relative number of positive and negative elements in the given column. Figure 8, facing, typifies the various combinations of units which may be used to generate all the necessary outputs required by the program.

Since the solution to both problems outlined above involves the addition of amplifiers to integrator outputs, it was convenient and economical to consider them simultaneously in the automatic program. Figure A-6 (Part 1) of the program for Integrator Loading Alleviation and Amplifier Assignment shown in Appendix A serves to generate and store data which specifies for each column of the matrix (1) the number of positive elements; (2) the number of negative elements; and (3) the total number of non-zero elements. Part 2 of the same program utilizes this information to determine the simplest integrator-amplifier combinations (from those available) that will provide the necessary positive and negative outputs without overloading the integrators and/or amplifiers. This is accomplished by attempting to assign the simplest combination first. If this does not result in the necessary outputs, the next simplest combination is assigned, etc. The results of these decisions are stored in the form of print statements which explicitly indicate the units that are to be wired together.

<u>Equipment Selection and Wiring Specification</u>

While it would be desirable to develop a general method for assigning integrators, amplifiers, potentiometers, and interconnections, such assignments are of necessity closely related to the particular analog computer for which the problems are prepared. For this reason the discussion of equipment selection and the resulting wiring requirements will be orientated to the Reeves computer used as the basis for the example digital computer program. It is anticipated that this discussion will provide an insight to the problems to be solved and a basis for more sophisticated routines in a more general program.

Before considering the development of the procedure for selecting the analog equipment and specifying the wiring connections, it is necessary to consider some of the features peculiar to the Reac Model 101. The computer is arranged in two bays (A and B), each with seven integrators (numbered A1. A7 and B1..B7), 23 potentiometers (numbered A1...A23 and B1...B23), seven summing amplifiers (numbered A8...A14 and B8...B14), and six inverting amplifiers (numbered A15...A20 and B15...B20). The two bays may be interconnected by means of 44 interconnections (numbered 101...122 and 201...222).

The problem of equipment assignment, therefore, reduces to one of selecting, from the available equipment, the units to be assigned in the solution of a given problem, and to specify the necessary patchboard connections to interconnect the units for the problem solution. (Note that the required amplifiers were assigned by the preceding routine).

Integrator Assignment. The integrators, starting with A1, are assigned to the problem variables in direct correspondence with the order in which they occur in the matrix equation. That is, integrators A1 through A7 are assigned to variables X_1 thru X_7 and integrators B1 through B7 are assigned to variables X_8 through X_{14}. In the case of systems of less than fourteenth order, no connections are specified to the integrators reserved for the unused higher-order variables. This simple system for the assignment of integrators to problem variables has the advantage that it results in an easily-remembered correspondence between problem variable and integrator. This relationship is maintained for all problems set up by the automatic system.

Potentiometer Assignment. A review of the input matrix equation to the digital computer will show that there is no provision for indicating which coefficients the analog computer operator might desire to change during the course of the investigation. A potentiometer was therefore included in the implementation of the gain associated with each non-zero element of the coefficient matrix. Such a procedure will, in some cases, result in potentiometer settings of 'one', but the additional flexibility achieved justifies the extra equipment. Further potentiometers were used on only one bay of the equipment in problems where this was possible (problems with not over 23 non-zero elements in the coefficient matrix). The ease of setting potentiometers on only one bay justifies the additional interconnections required by this procedure. In cases where it was necessary to use potentiometers from both bays of the equipment, they were selected so as to reduce the number of interconnections by assigning them (where possible) from the same bay as the units providing their input.

These decisions are inherent in the first portion of the Equipment Selection and Wiring Specification program shown in Appendix A (Fig. A-7, Part 1). This portion of the program identifies the potentiometers to be used with each non-zero element of the coefficient matrix by developing print instructions which specify the connections from the proper integrator, or its associated amplifier output, to the potentiometer input. The program also generates print instructions which inform the operator of the correspondence between a given potentiometer and the element of the coefficient matrix to which it is related.

Input Gain Specifications. The print instructions referred to in the above discussion associate each non-zero element in the coefficient matrix with a specific potentiometer on the analog computer. It then becomes necessary for the automatic program to select the amplifier gain to which the potentiometer should be connected. This is accomplished by the second part of the Equipment Selection and Wiring Specification program shown in Appendix A (Fig. A-7, Part 11). The selection is made by considering the magnitude of the matrix element with which the potentiometer is associated, in conjunction with the gains available for assignment to the potentiometer.

Potentiometer Setting. Once the selection of the integrator gain to which a potentiometer is to be connected has been made, it is possible to calculate the setting for that potentiometer. This calculation is performed by the part of the Equipment Selection and Wiring Specification program of Appendix A (Fig. A-7, Part 11). The calculation is made by using the relation

$$\text{Pot setting} = E_{ij} / \text{ integrator gain} \qquad (2\text{-}35)$$

The result of the calculation is stored in the form of a print statement which specifies the potentiometer setting for the subsequent use of the analog computer operator.

Integrator Input Connections. The portion of the automatic program discussed above specifies the connections from the integrator outputs to the potentiometer inputs, selects the integrator gain to which the potentiometer is to be connected, and calculates the potentiometer setting. The remaining part of the Equipment Selection and Wiring Specification program generates the print statements which specify the connections from the potentiometer outputs to the integrator inputs at the appropriate gain.

Output Program

The instructions generated by the program to this point are stored in the memory of the digital computer. The remaining task of the automatic program is to print out the wiring instructions, potentiometer settings, scale factors, and correspondences between potentiometers and matrix elements. The details of how this is to be accomplished are left to the digital computer programer and therefore do not properly form a part of this report. The final results of the output routine are presented in the sample problem shown in Appendix B.

III. Use of the Automatic Program

The usefulness of the automatic program described in Chapter III may best be indicated by a consideration of the operations required to solve a problem using the program. The operations which the analog computer operator must perform are to prepare the problem for the automatic program and apply its output to complete the analog computer solution.

Problem Restrictions

Before proceeding with the discussion of the use of the automatic program, it is necessary to review the limitations on the input problem which result from the assumptions made in the development of the program. First, the program is designed to handle only systems of linear homogeneous differential equations with constant coefficients. Second, systems of equations requiring not more than 14 integrations may be handled provided not more than seven inputs are required to any integrator and/or not more than seven outputs are required from any integrator. Third, the system of equations must require no more than 46 potentiometers.

Problem Formulation

Problems meeting the above restrictions may be prepared for analog computer solution by the automatic program. To use the program, the operator must convert the input equations to the standard matrix form specified by Equation (2-1). An example of this conversion procedure is illustrated in Figure 5. The coefficient matrix of the system of equations in standard form together with the maximum value of each of the variables (of the equations in standard form) provide the input to the digital computer.

It may be noted that the digital computer requires that the coefficients of the dependent variables be expressed in numerical, not general, terms. The resulting analog computer circuit and potentiometer settings, therefore, are for only one problem condition. However, once a problem has been programed by the digital computer, it is possible to investigate it at other than the original gains, provided that the maximum gain condition was used in setting up the problem. Therefore, problems to be investigated at several gain conditions should be presented to the automatic program with the maximum gains specified.

It may be observed that the analog computer used in this study has initial condition potentiometers permanently wired to the integrators. Therefore, the automatic program is not concerned with the initial conditions of the problem variables. The operator, on the other hand (as is shown in the sample problem, Appendix B), must scale the initial conditions by using the scale factors selected by the automatic program.

Use of the Automatic Program Output

As has been noted earlier, the output of the digital program is in the form of printed statements which specify the analog computer wiring, potentiometer settings, variable scale factors, and correspondence between potentiometer and coefficient matrix elements. These printed instructions may then be used to draw the conventional analog computer wiring diagram. However, this intermediate step is not necessary. The computer patchboard may be wired, the potentiometers set, and the problem run by using the printed instructions directly.

It will be noted that the automatic program takes no note of any wiring connections from the analog computer units to recorder inputs. The reason is, of course, that since the equations in matrix form include a variable for each integration in the problem, it is usually not necessary to record all of the problem variables. However, each integrator in the problem has an unused output connection as a result of the provisions made for loading alleviation. Since the recorder input imposes no significant load on the integrator output, the operator simply chooses the variables he wishes to record and patches from the appropriate integrator to the recorder.

The final task of the operator is scaling the output recordings of the analog computer. The time base of the recordings may be scaled to real time by the use of the time scale factor. The amplitudes of the output variables may be scaled by the application of the amplitude scale factors.

IV. Conclusions and Recommendations

The principles which have been discussed at length in the preceding pages (and which have been implemented in the program exhibited in Appendix A) attest to the feasibility of using a digital computer to assist in the programing of an analog computer. Within the framework of the stated limitations and assumptions, it has been demonstrated that the decisions, calculations, and operations usually required of an analog computer operator in the preparation of a problem for analog solution can be accomplished automatically with reasonable success. Very briefly, these include the determination of the need for, and the calculation of, appropriate amplitude- and time-scale factors; the selection and arrangement of integrators, amplifiers, and potentiometers into combinations which will provide the necessary outputs and inputs without overloading the units; and the calculation of potentiometer settings.

Comparison with Optimum Program

On the basis of tests to which the automatic program has been subjected it can be stated that the program will generate a correct analog computer circuit for any problem within the specified limitations. It should be noted, however, that the solution produced by the automatic program may not, in certain circumstances, be the most economical in terms of the use of analog computer equipment. The number of integrators used in any case is unique, since the number of integrations required is specified by the order of the input system of equations. On the other hand, the number of potentiometers specified by the automatic program for a given problem may be greater than a manual solution would indicate. This disparity arises because of the method of specifying the input equations. When the original problem is a system of equations of higher order than the first, the conversion to a system of first-order equations creates 'artificial' variables for each derivative, i.e.,

$$X_1 = Y$$
$$X_2 = -\dot{X}_1 = -\dot{Y}$$
$$X_3 = -\dot{X}_2 = Y$$

Therefore, this introduces exactly k potentiometers which are unnecessary, where k equals the number of derivatives higher than first-order appearing in the original problem (see Example Problem, Appendix B).

Therefore, the only other undesirable difference in the analog computer circuit that may occur between the automatic and manual programs is that the former may prescribe a greater number of inverting amplifiers. It will be recalled that the choice of amplifiers to be added to the integrators is based on the need to minimize loading effects at the integrator outputs and to form the negative outputs specified by negative matrix elements. A systematic scheme for reducing to a minimum the number of amplifiers which would be needed to solve a given problem has been developed for a more general program (Appendix C). However, this technique is based on the assumption that integrator loading effects are non-existent. This implies, of course, that in the extreme case, if all the elements of the coefficient matrix could be made positive, no additional amplifiers would be needed. Obviously, a scheme such as this could not be used in the present program, where loading effects limit the number of direct usable integrator outputs to a maximum of four. Furthermore, since only 46 potentiometers and a total of 26 inverting amplifiers are available, the present program will never exhaust the available amplifiers.

Finally, it appears possible to develop a more comprehensive program which will assign the absolute minimum number of amplifiers required to solve a given problem, should this additional complexity appear justified.

Recommendations

At this point, the reader will undoubtedly have some ideas of his own for extending the program which has been described. The following suggestions and recommendations are presented in the hope that they will stimulate further interest and serve as a basis for modifying and expanding the automatic technique to more advanced and comprehensive systems. Some of the more important extensions and modifications to the present program would be as follows:

1. A technique should be formulated which would suppress those potentiometers introduced as a result of converting a higher-order system of equations to one of first order. This could be accomplished with a sub-routine in the present program. However, it would be more advantageous to be able to input the higher-order equations directly to the digital computer. This would be facilitated by (a) the formulation of a general symbology which could be used to express the input equations without prior conversion to standard matrix form; and (b) the inclusion

of a provision for handling the coefficients of the equations in general terms so that either a range of coefficient values (or several values) could be specified. The digital computer could then be programed to make the conversion to standard form and hence, to specify connections and potentiometer settings to handle the coefficient values specified. Such an input routine would also facilitate the program extension to include non-homogeneous linear differential equations and ultimately non-linear differential equations.

2. A procedure for generating the assignment of the absolute minimum number of inverting amplifiers. This extension might be based upon the discussion of minimization given in Appendix C. Another technique which might be used would be an iterative procedure which, after calculating a first solution, would try different alternatives, stopping only when no further improvement was possible.

3. The inclusion of an iterative procedure which could guarantee a selection of the analog computer components in such a manner that the minimum number of interconnections would be required.

Bibliography

1. Hohn, F. E. Elementary Matrix Algebra. New York: The Macmillian Company, 1958.

2. Honnell, P. M. and R. E. Horn. "Matrices in Analog Mathematical Machines." Journal of the Franklin Institute, 260:193-207 (1955).

3. ..... "Analog Computer Synthesis and Error Matrices." Transactions of the American Institute of Electrical Engineers, 75:22-32 (1956).

4. Johnson, C. L. Analog Computer Techniques. New York: McGraw-Hill Book Company, 1956.

5. Stein, M. L., and J. Rose. "Changing from Analog to Digital Programing by Digital Techniques." Journal of the Association for Computing Machinery, 7:10-23 (1960).

Appendix A

The Automatic Program

The automatic program illustrated in the following pages consists of a series of inter-related logic flow diagrams (Figure A- 1,7) which implement the principles and techniques developed and discussed in Chapter II. The complete program is logically sub-divided into five phases which include the (1) Problem Check Program, (2) Amplitude Scaling Program, (3) Time Scaling Program, (4) Integrator Loading Alleviation and Amplifier Assignment Program, and (5) Equipment Selection and Wiring Specification Program. The specific objectives of each program together with the procedures used to accomplish these objectives are summarized on a facing page at the start of each program. The Glossary, which includes the terms used in the program, is provided as a fold-out at the end of the report. It is recommended that the Glossary be used during reference to the program.

It may be noted that the checks made for violation of the restrictions on the original problem are indicated in the program as exits through which the problem may be stopped. However, a number of exits were included for use during the testing of the program which are not possible exits during its normal operation. These exits do not, therefore, imply the existence of any further restrictions on the problem.

Problem Check

(Program Summary)

Objectives:

(1) Provide a means of rejecting a given problem in cases where:
  (a) More than seven (7) input to any integrators are required.
  (b) More than 46 potentiometers are required.

(2) Determine and store, in memory, the absolute values of the maximum and minimum (non-zero) matrix elements for future use.

Procedures:

(1) Count the number of non-zero elements in each row of the coefficient matrix [ΣAUM (I)] and reject the problem if the count totals more than seven for any row.

(2) Count the number of non-zero elements in the matrix (TAUM) and reject the problem if the total is greater than 46.

(3) Record the magnitudes of the largest (AMAX) and smallest (AMIN) non-zero elements in the matrix.

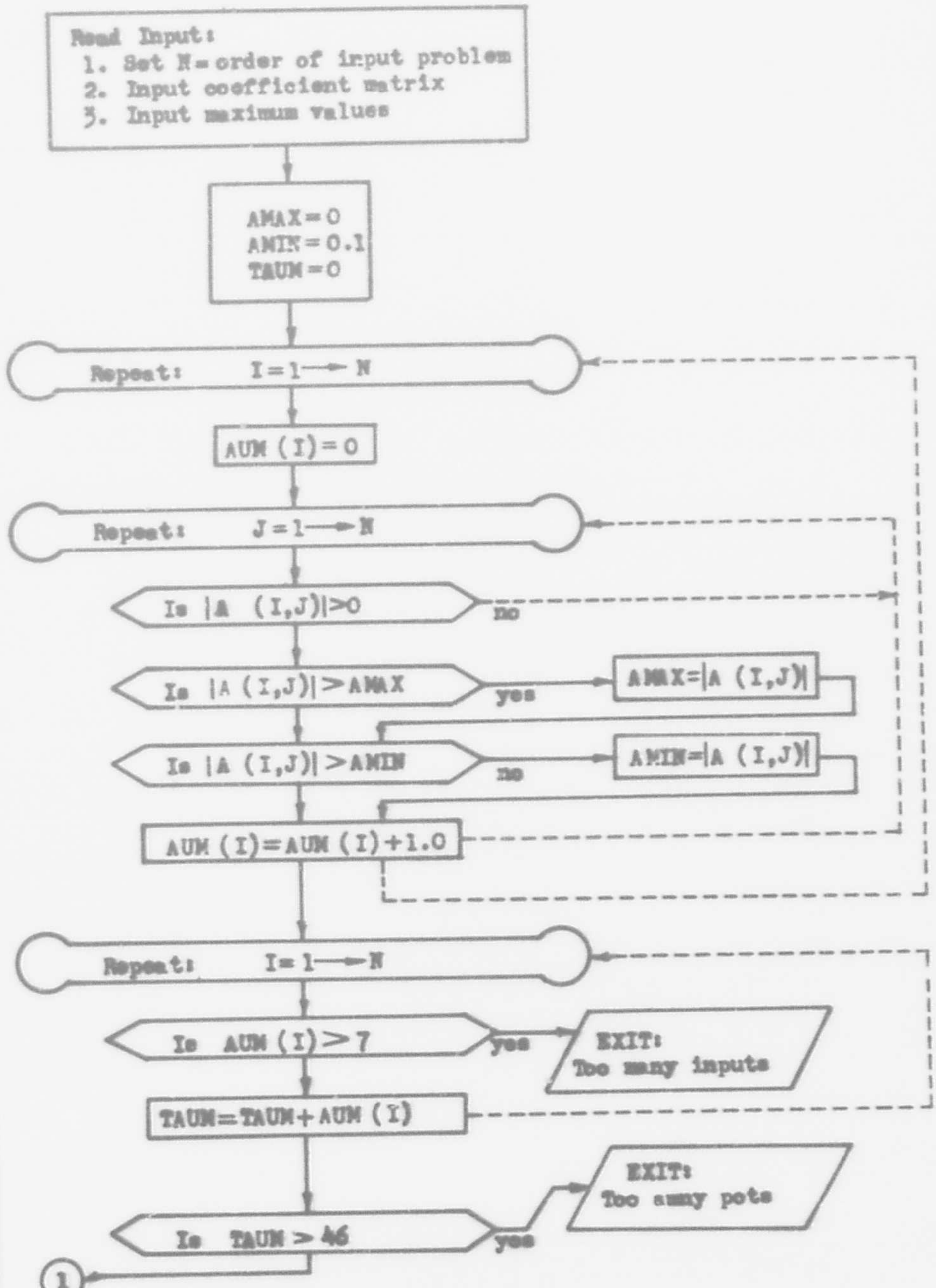


Figure A-1
Problem Check Program

Amplitude Scaling

(Program Summary)

Objectives:

(1) Provide a means for generating amplitude scale factors for the dependent variables from the known maximum values and maximum permissible outputs from the analog computer units.

(2) Provide a means for forming the amplitude-scaled matrix from the scale factors and the input coefficient matrix.

Procedures:

(1) Calculate an amplitude-scale factor [S (I)] for each row of the coefficient matrix (A).

(2) Round-off the calculated amplitude-scale factors to facilitate analog recorder interpretation.

(3) Generate the amplitude-scaled matrix (E') from the scale factors and the input coefficient matrix.

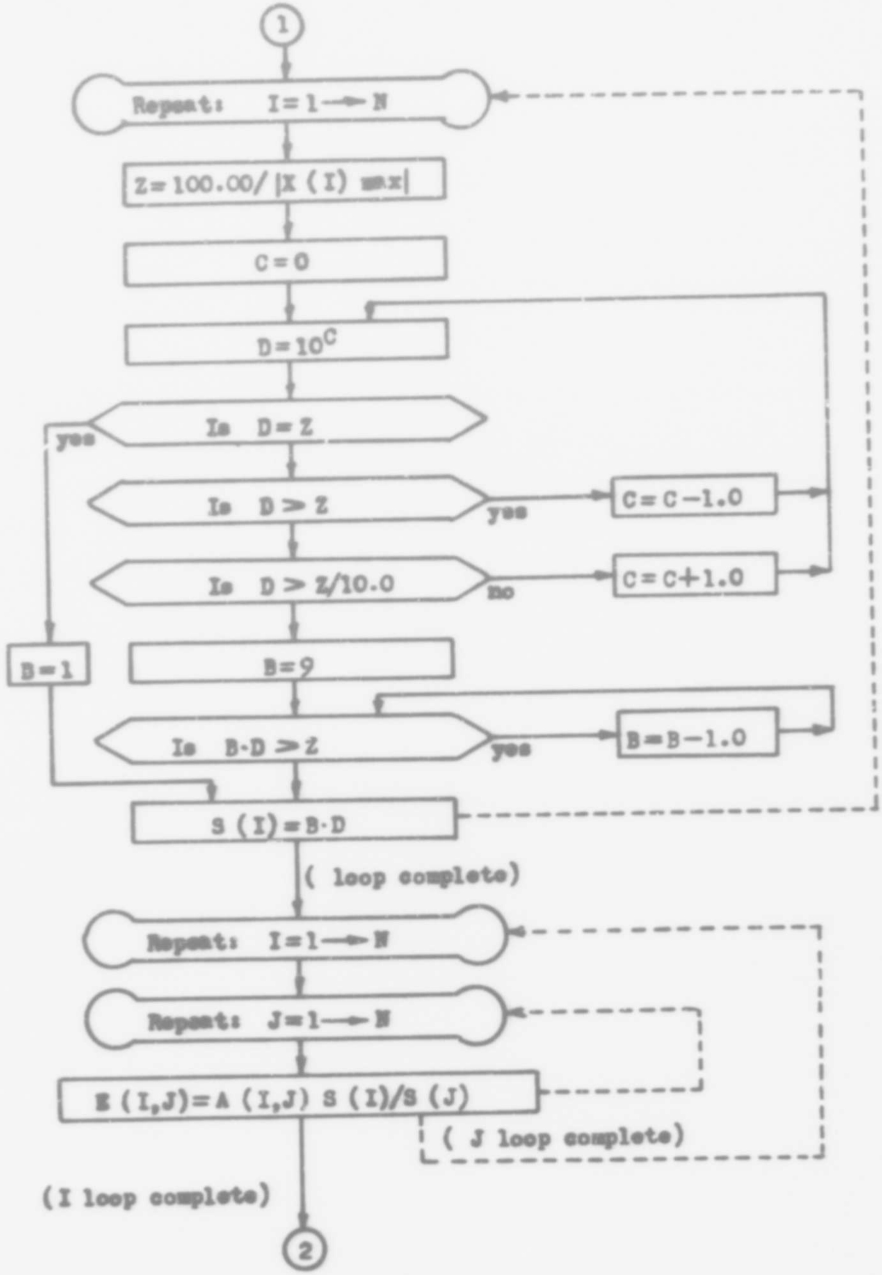


Figure A-2

Amplitude Scaling Program

Time Scaling (Part I)

(Program Summary)

Objectives:

Develop a procedure which can be applied to the amplitude-scaled matrix so as to arrange the elements of each row in descending order of magnitude.

Procedures:

(1) Generate a new matrix (G) from the amplitude-scaled matrix (E') such that:
   (a) G (I,J), the general element of the new matrix, is a number which is equal to the column subscript (J) of the element of the E' matrix from which it was formed.
   (b) The subscripts of the G matrix indicate the row, and the order within the row, respectively, of the elements E (I,J) in descending order of magnitude.

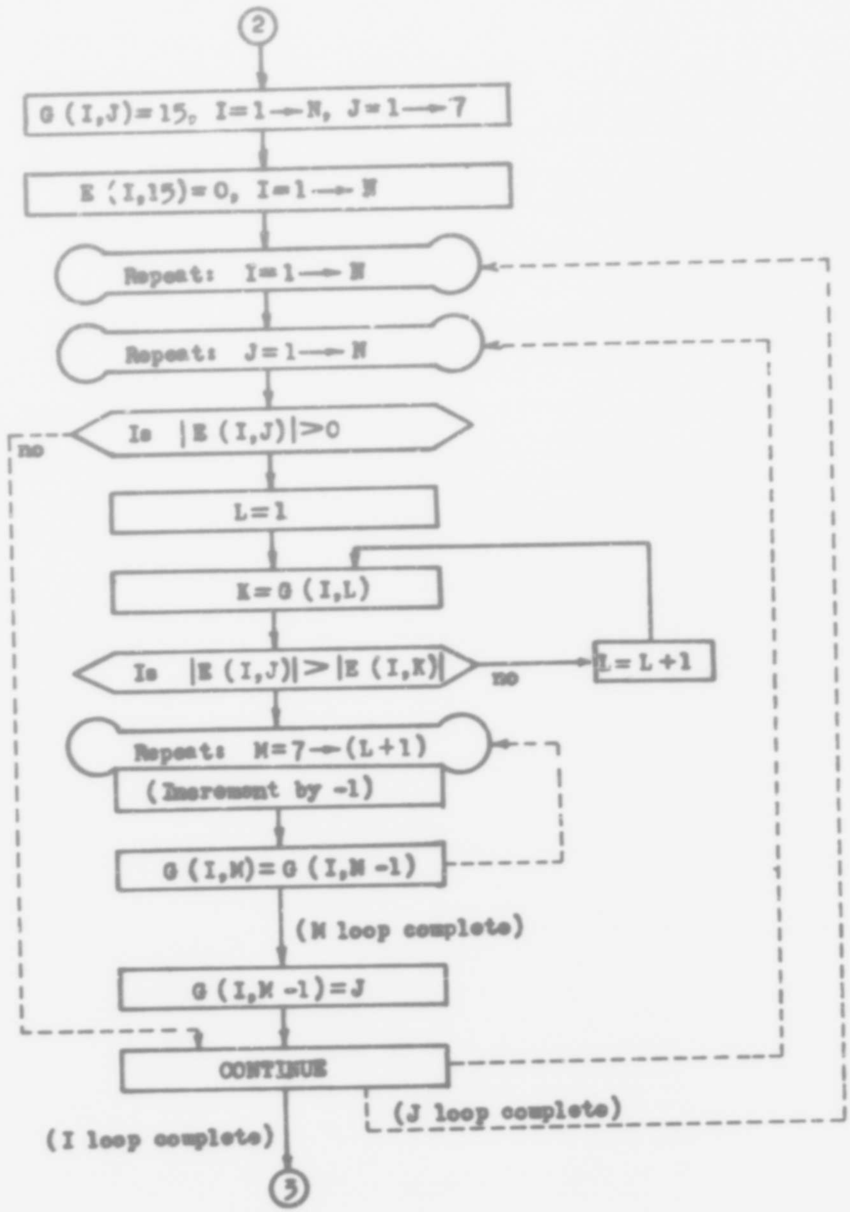


Figure A-3

Time Scaling Program (Part 1)

Time Scaling (Part II)

(Program Summary)

Objectives:

Provide a means for:

(1) Time scaling the amplitude-scaled matrix so that machine problem variables remain proportional to the original problem variables.

(2) Comparing the inputs specified by the elements of each row of the scaled matrix with the integrator gains available to form the inputs -- and rescale if necessary.

Procedures:

(1) Determine the necessity for time scaling.

(2) Calculate the time-scale factor (T_1).

(3) Utilize the G matrix generated in Part I of the program to determine the compatibility of the inputs required (after scaling) with the integrator gains available for each row of the scaled matrix, and rescale until a satisfactory time-scale factor is determined.

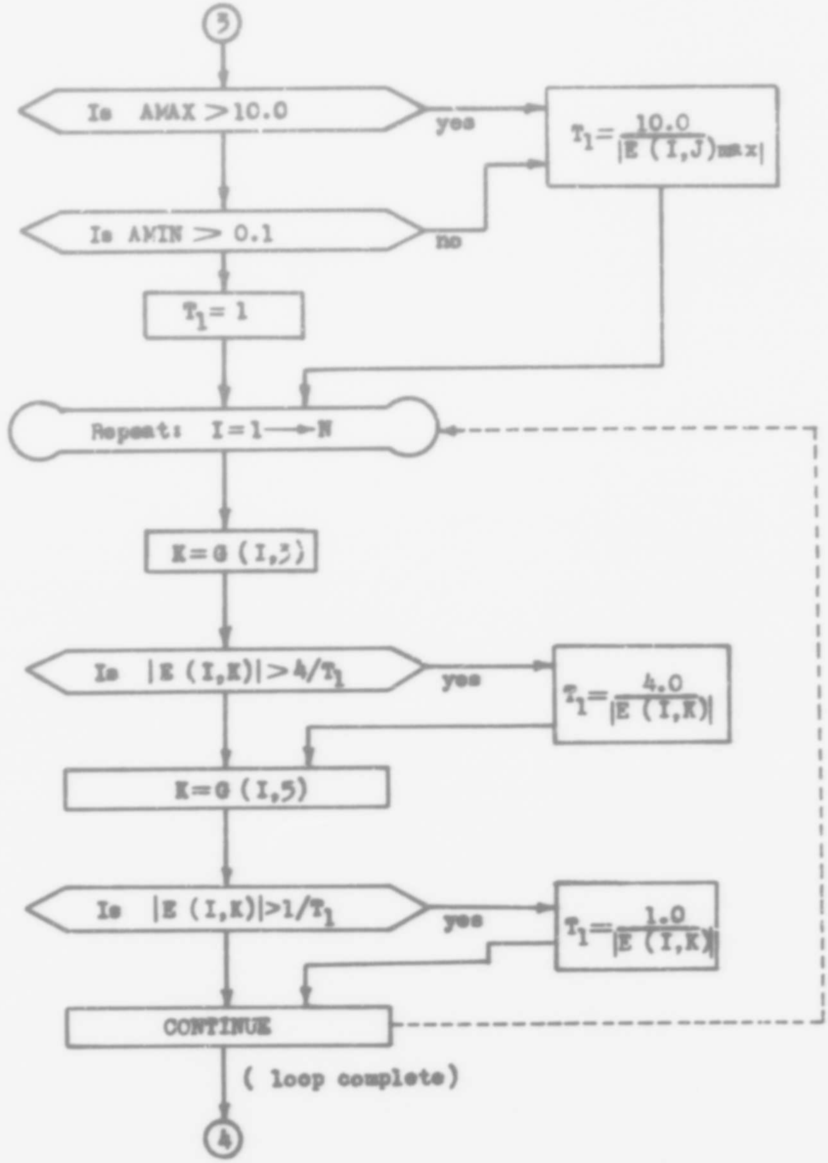


Figure A-4

Time Sealing Program (Part II)

Time Scaling (Part III)

(Program Summary)

Objectives:

Provide means for:

(1) Modifying the time-scale factor calculated in Part II for ease in analog recorder interpretation.

(2) Forming the time-and amplitude-scaled matrix.

Procedures:

(1) Round-off the scale factor (T_1) calculated in Part II and generate the optimum time-scale factor (T).

(2) Calculate the time-and amplitude-scaled matrix ($E=E'T=TSAS^{-1}$).

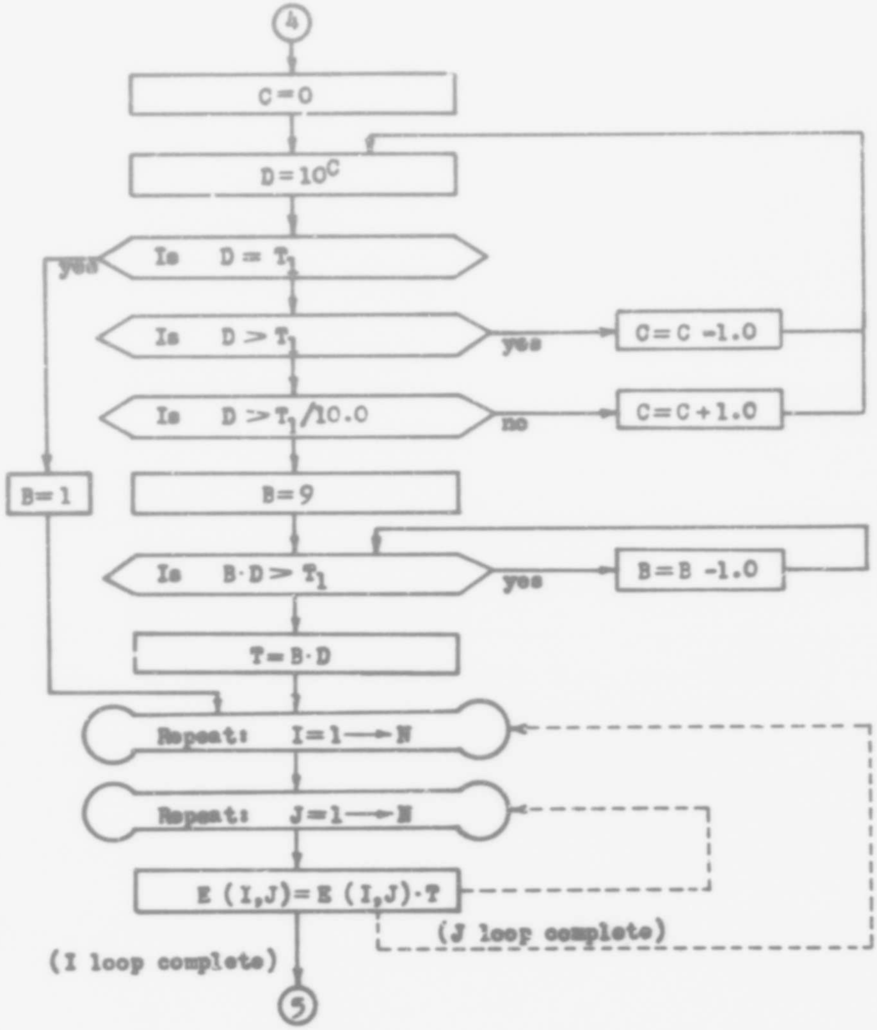


Figure A-5

Time Scaling Program (Part III)

Integrator Loading Alleviation

and Amplifier Assignment

(Program Summary)

Objectives:

Provide a means to:
(1) Eliminate integrator and/or amplifier loading effects.
(2) Select and specify the simplest integrator-amplifier combination which will provide the necessary positive and negative outputs.
(3) Reject the problem if too many outputs are required.

Procedure:

(1) Examine each column of the scaled coefficient matrix (E) and:
  (a) Count the number of positive elements OKF (J).
  (b) Count the number of negative elements CKN (J).
  (c) Count the total number of non-zero elements OK (J).
(2) Select the simplest integrator- amplifier combination which will satisfy the requirements of (1) above.
(3) Generate print statements which specify the wiring connections needed to combine the added units.
(4) Generate internal bookkeeping information for use in subsequent routines.

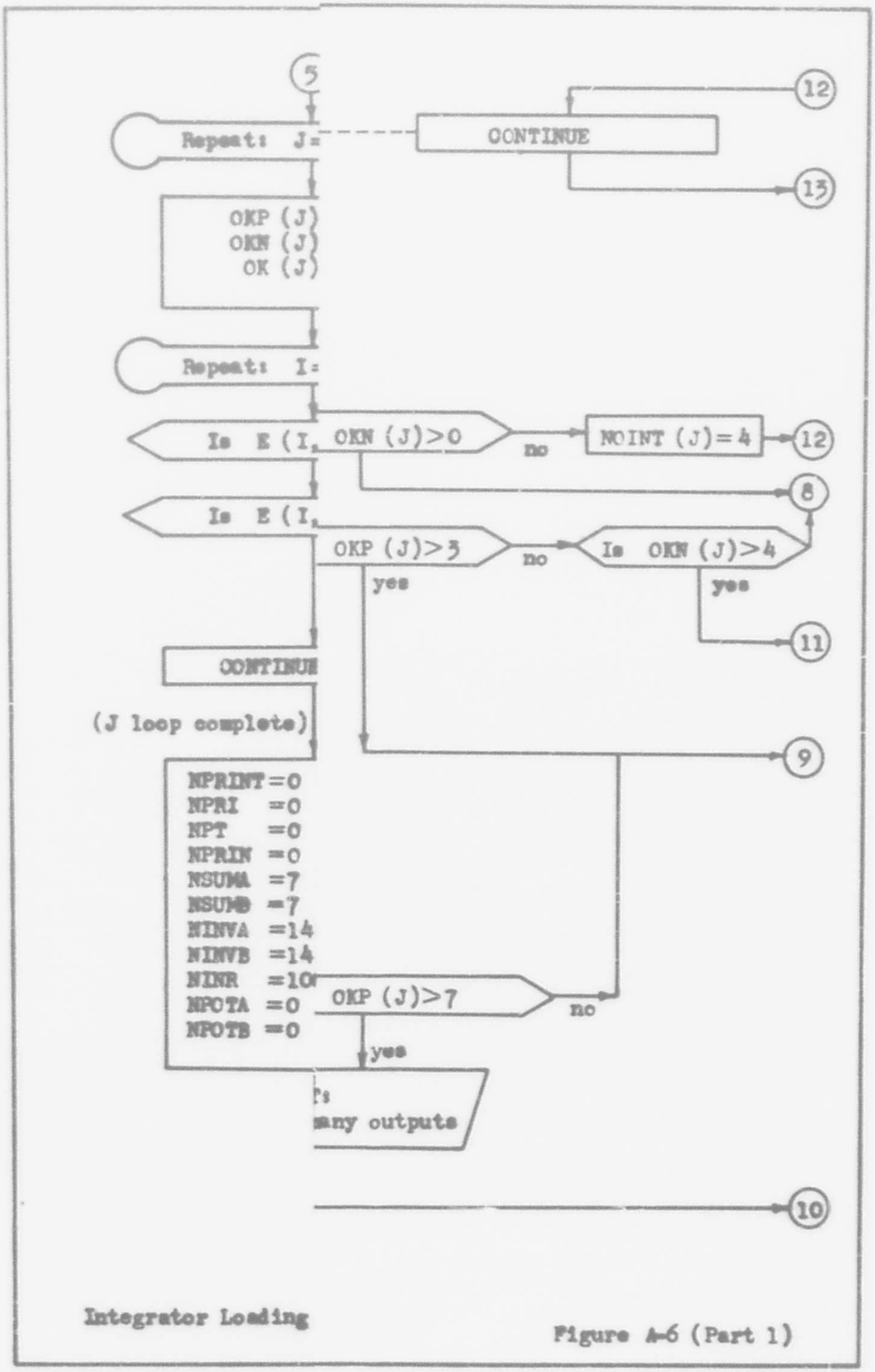


Figure A-6 (Part 1)

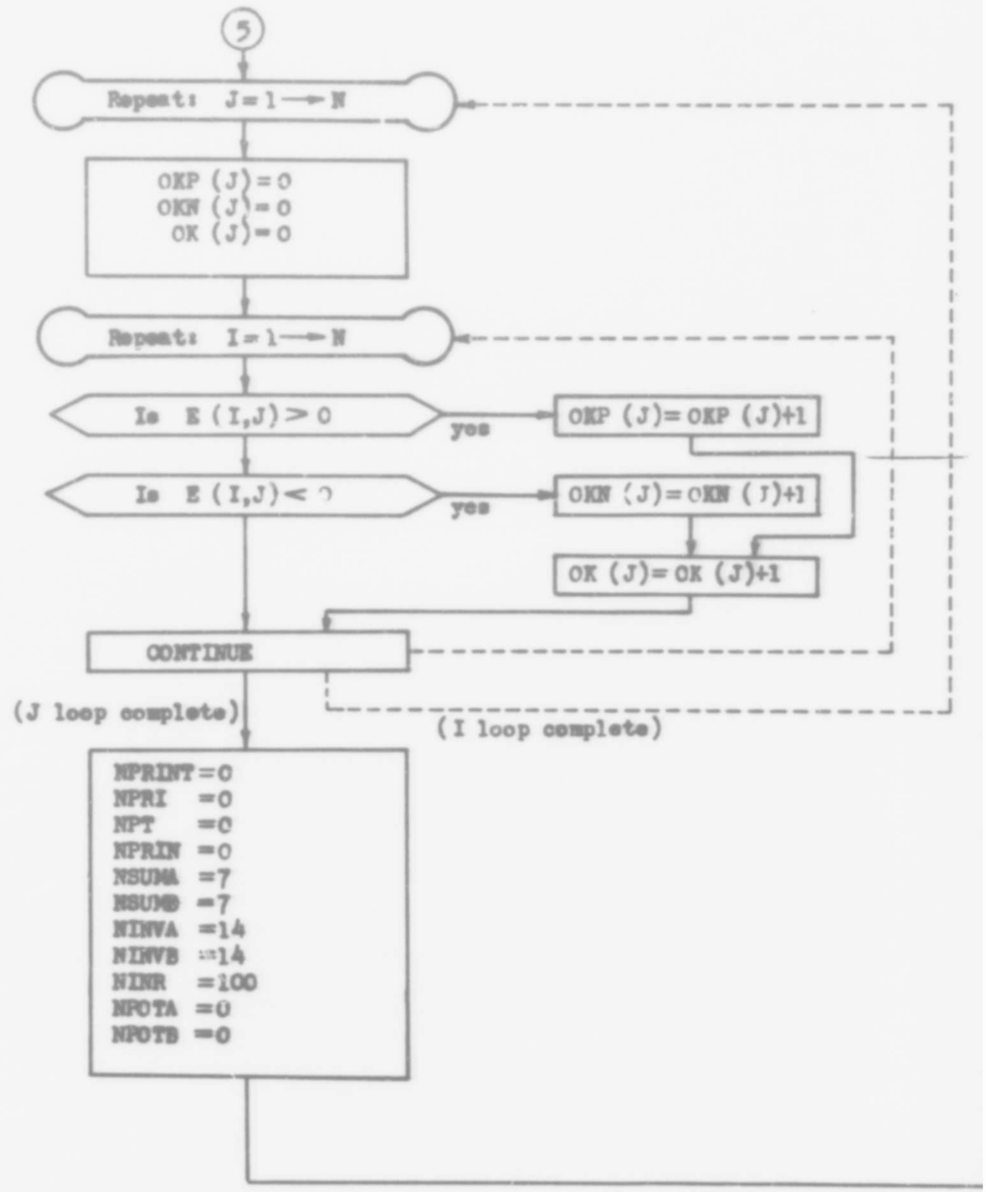


Figure A-6 (Part 1)

Integrator Loading Alleviation and Amplifier Assignment Program

1

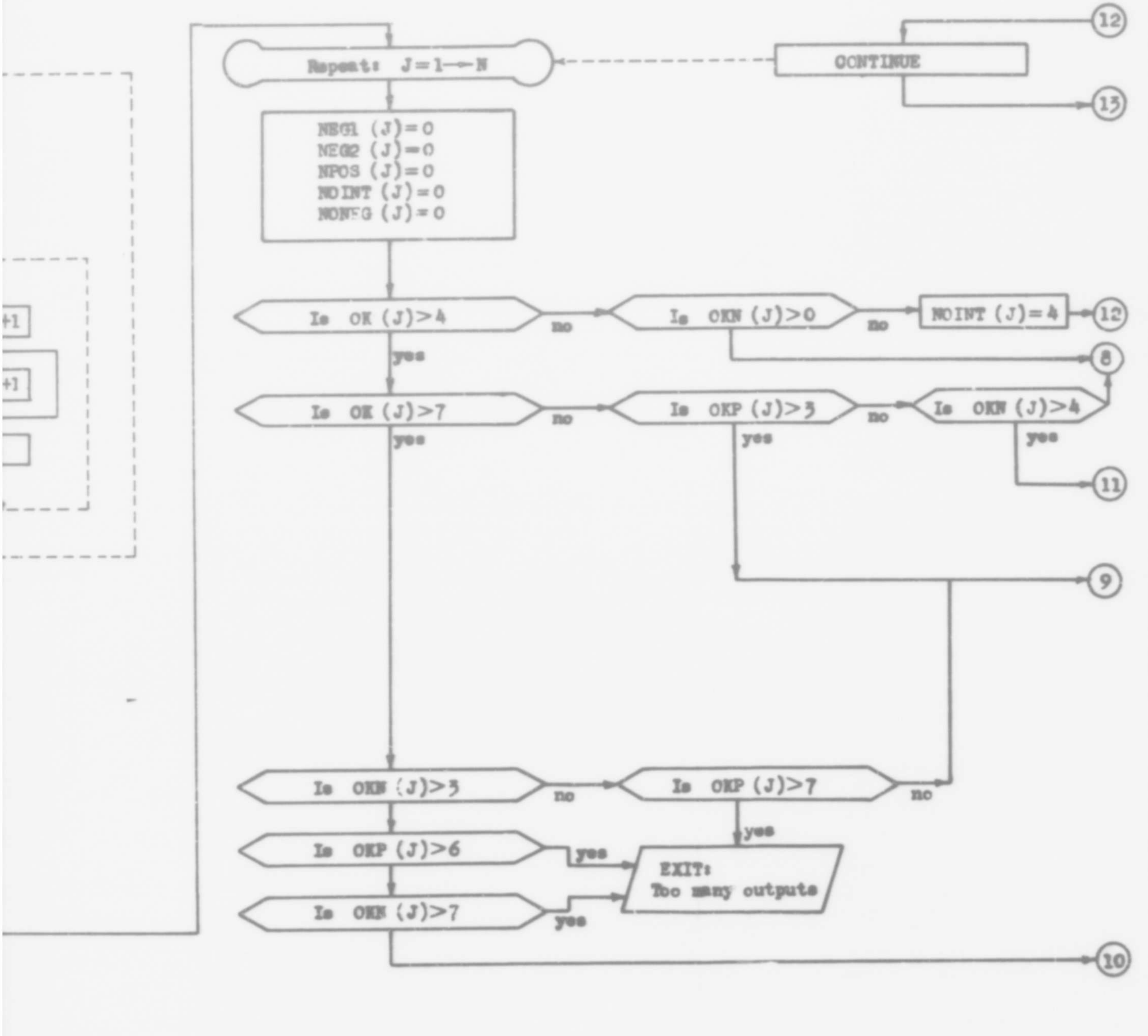
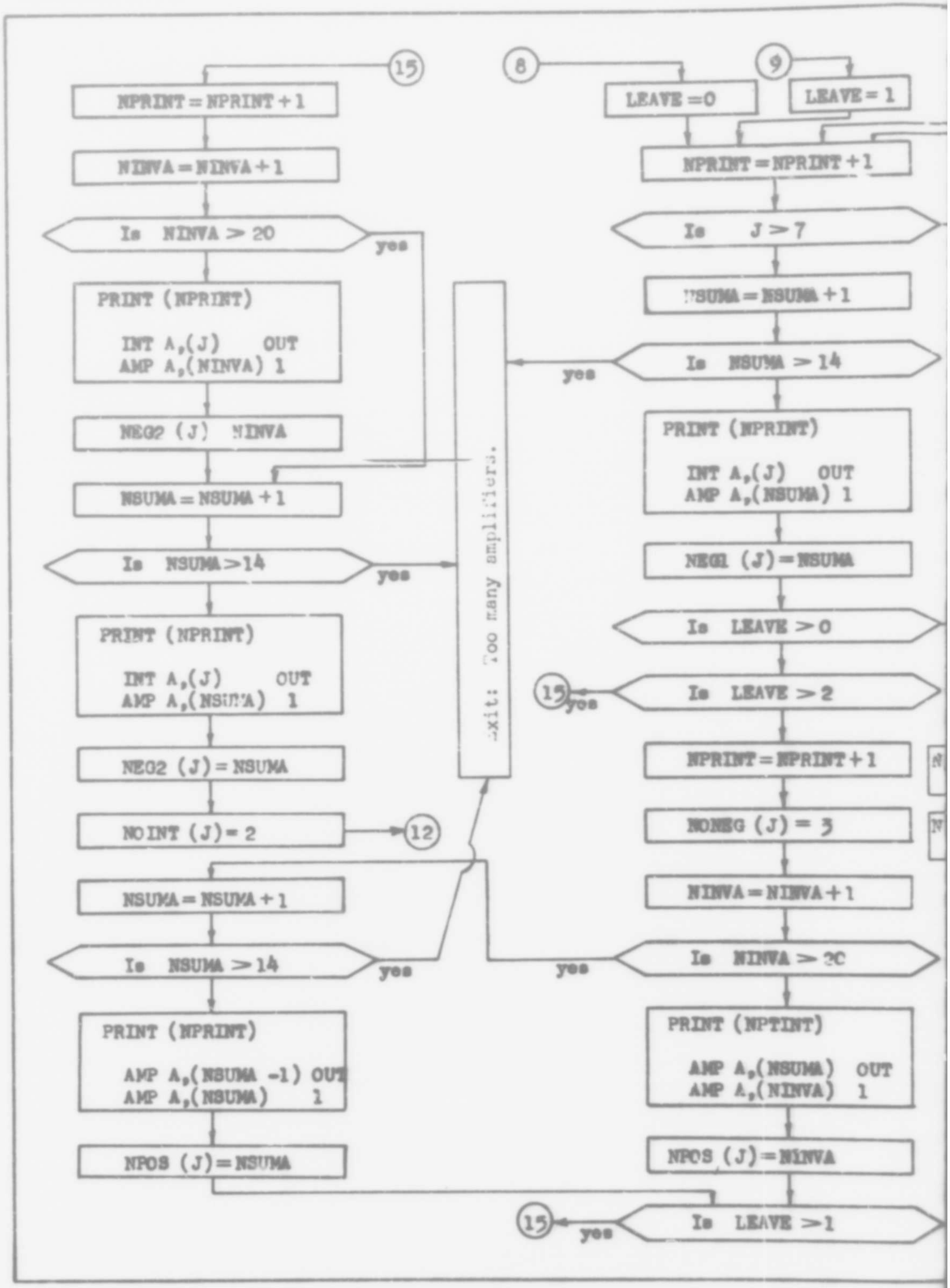


Figure A-6 (Part 1)


NPRINT = NPRINT + 1
NINVA = NINVA + 1
Is NINVA > 20
yes
PRINT (NPRINT)
INT A,(J) OUT
AMP A,(NINVA) 1
NEG2 (J) NINVA
NSUMA = NSUMA + 1
Is NSUMA > 14
yes
PRINT (NPRINT)
INT A,(J) OUT
AMP A,(NSUMA) 1
NEG2 (J) = NSUMA
NOINT (J) = 2
12
NSUMA = NSUMA + 1
Is NSUMA > 14
yes
PRINT (NPRINT)
AMP A,(NSUMA -1) OUT
AMP A,(NSUMA) 1
NPOS (J) = NSUMA
Exit: Too many amplifiers.
15
8
9
LEAVE = 0
LEAVE = 1
NPRINT = NPRINT + 1
Is J > 7
NSUMA = NSUMA + 1
Is NSUMA > 14
yes
PRINT (NPRINT)
INT A,(J) OUT
AMP A,(NSUMA) 1
NEG1 (J) = NSUMA
Is LEAVE > 0
Is LEAVE > 2
yes
NPRINT = NPRINT + 1
NONEG (J) = 3
NINVA = NINVA + 1
Is NINVA > 20
yes
PRINT (NPTINT)
AMP A,(NSUMA) OUT
AMP A,(NINVA) 1
NPOS (J) = NINVA
Is LEAVE > 1
yes

1

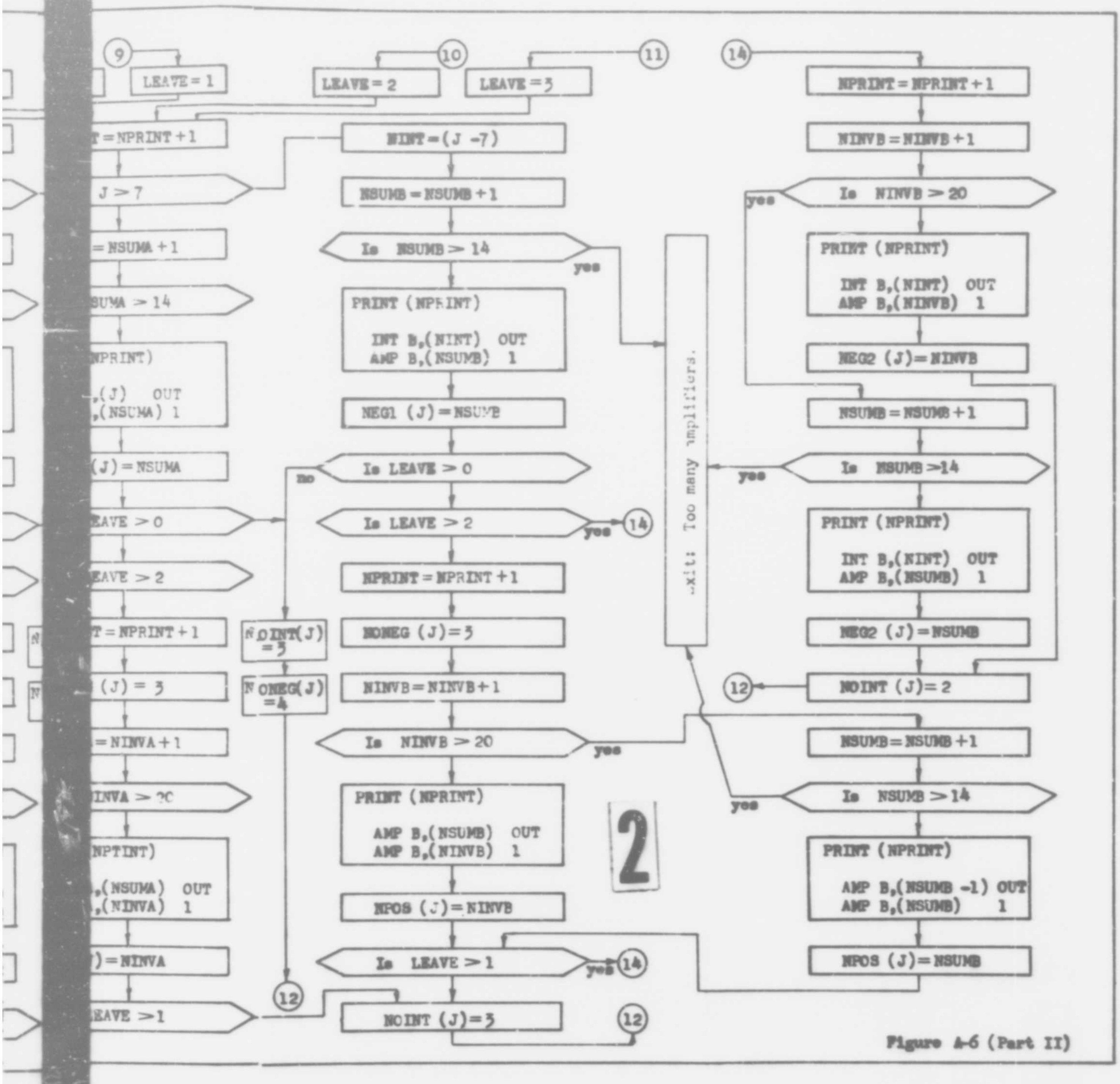
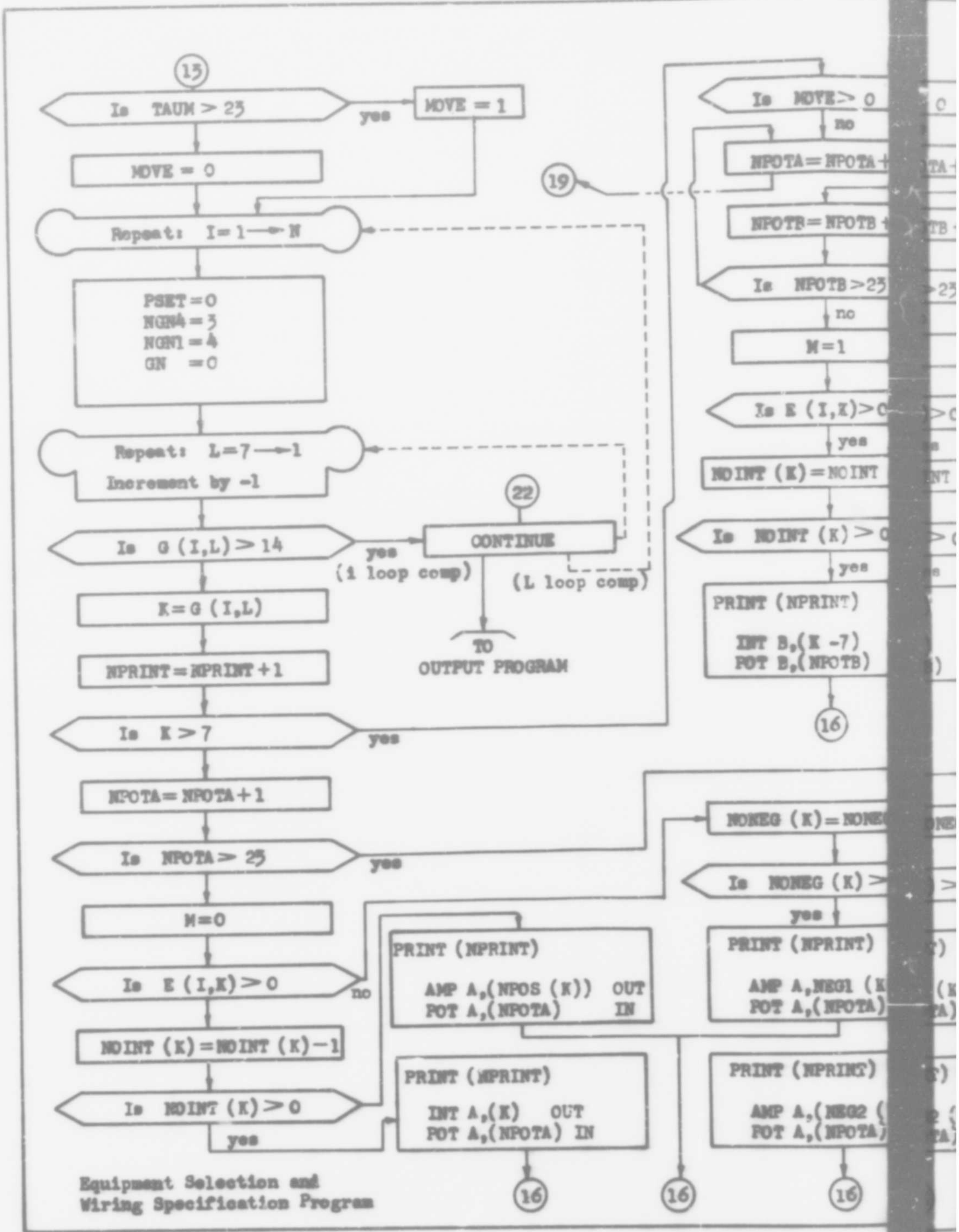


Figure A-6 (Part II)



15
Is TAUM > 23
yes
MOVE = 1
MOVE = 0
Repeat: I=1 → N
PSET = 0
NGN4 = 3
NGN1 = 4
GN = 0
Repeat: L=7 → 1
Increment by -1
22
Is G (I,L) > 14
yes
CONTINUE
(i loop comp)
(L loop comp)
TO
OUTPUT PROGRAM
K = G (I,L)
NPRINT = NPRINT + 1
Is K > 7
yes
NPOTA = NPOTA + 1
Is NPOTA > 23
yes
M = 0
Is E (I,K) > 0
no
NOINT (K) = NOINT (K) − 1
Is NOINT (K) > 0
yes
PRINT (NPRINT)
AMP A,(NPOS (K)) OUT
POT A,(NPOTA) IN
PRINT (NPRINT)
INT A,(K) OUT
POT A,(NPOTA) IN
16
16
19
Is MOVE > 0
no
NPOTA = NPOTA +
NPOTB = NPOTB +
Is NPOTB > 23
no
M = 1
Is E (I,K) > 0
yes
NOINT (K) = NOINT
Is NOINT (K) > 0
yes
PRINT (NPRINT)
INT B,(K -7)
POT B,(NPOTB)
16
NONEG (K) = NONEG
Is NONEG (K) >
yes
PRINT (NPRINT)
AMP A,NEG1 (K
POT A,(NPOTA)
PRINT (NPRINT)
AMP A,(NEG2 (
POT A,(NPOTA)
16
Equipment Selection and
Wiring Specification Program

1

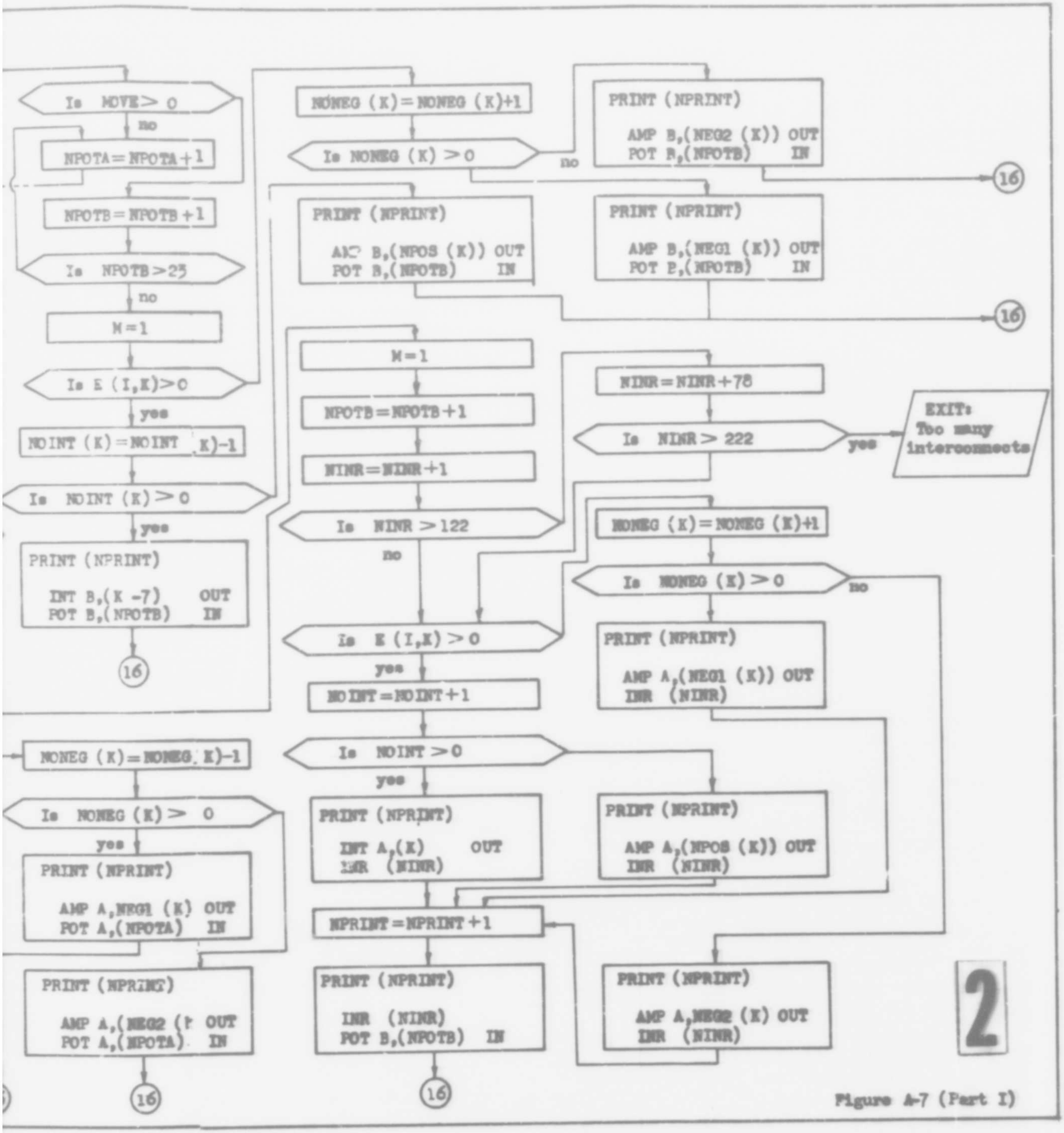
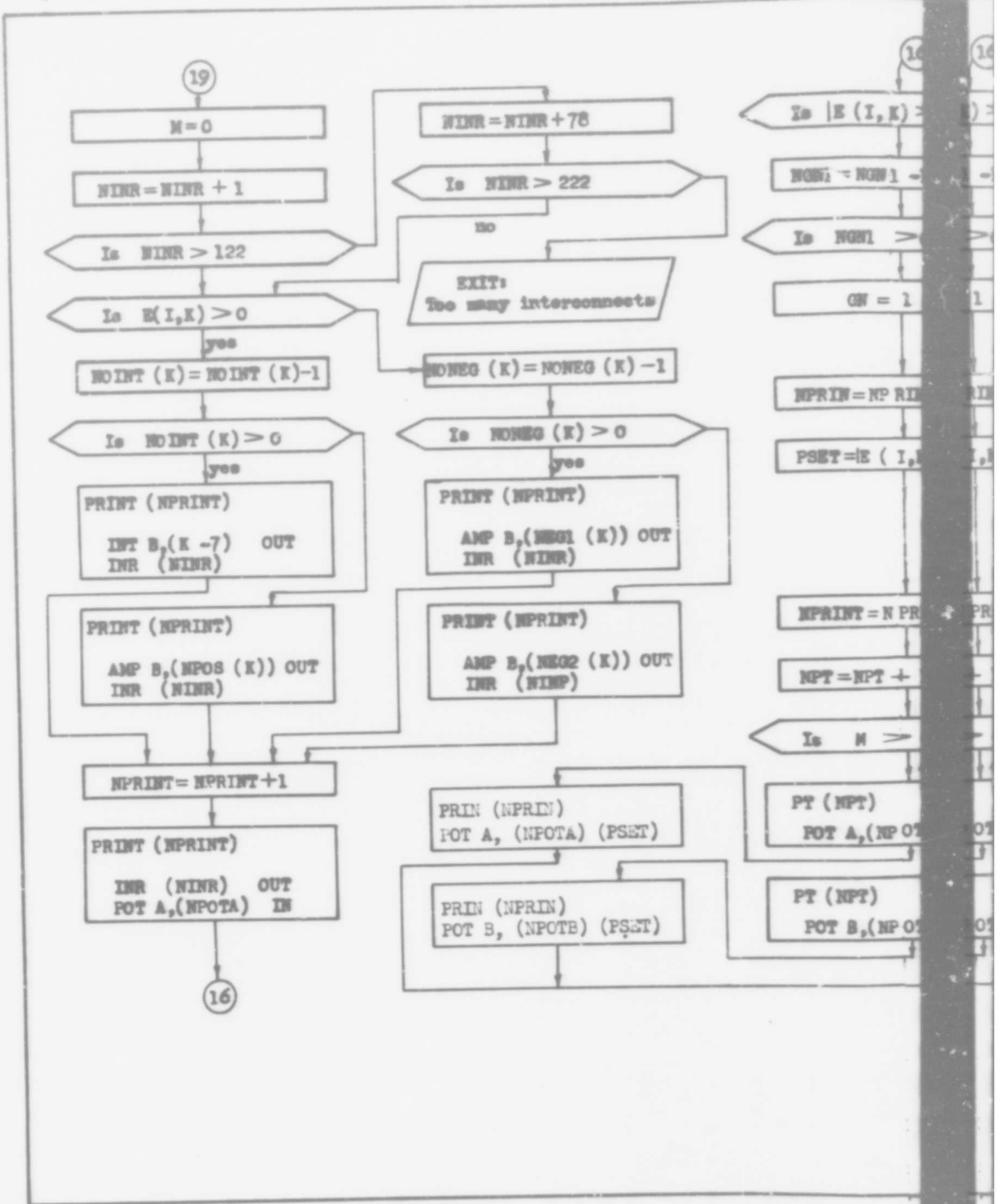


Figure A-7 (Part I)


19
M = 0
NINR = NINR + 1
Is NINR > 122
Is E(I,K) > 0
yes
NOINT (K) = NOINT (K)-1
Is NOINT (K) > 0
yes
PRINT (NPRINT)
INT B,(K -7) OUT
INR (NINR)
PRINT (NPRINT)
AMP B,(NPOS (K)) OUT
INR (NINR)
NPRINT = NPRINT + 1
PRINT (NPRINT)
INR (NINR) OUT
POT A,(NPOTA) IN
16
NINR = NINR + 78
Is NINR > 222
no
EXIT:
Too many interconnects
NONEG (K) = NONEG (K) -1
Is NONEG (K) > 0
yes
PRINT (NPRINT)
AMP B,(NEG1 (K)) OUT
INR (NINR)
PRINT (NPRINT)
AMP B,(NEG2 (K)) OUT
INR (NINP)
PRIN (NPRIN)
POT A, (NPOTA) (PSET)
PRIN (NPRIN)
POT B, (NPOTB) (PSET)
Is NGN1
GN = 1
PT (NPT)
PT (NPT)

1

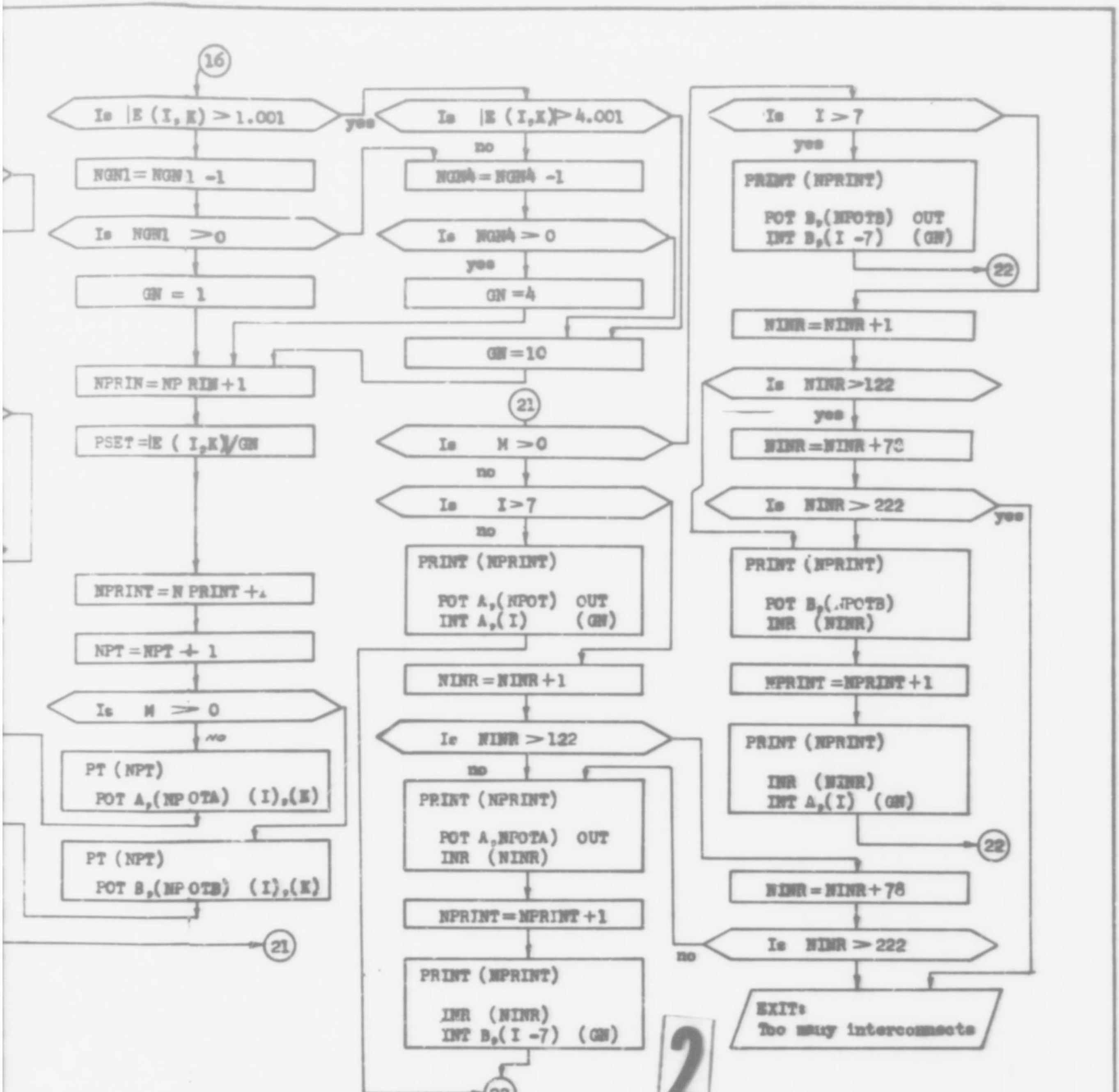


Figure A-7 (Part II)

Appendix B

Application of the Automatic Program to an Example Problem

To illustrate the results of the application of the automatic program for which principles were developed in Chapter II and which was presented in detail in Appendix A, an example problem will be programmed for analog computer solution. Consider the following set of coupled equations together with the maximum values indicated.

$$13.78\dot{u} + 0.088u + 0.392v + 0.741w = 0 \qquad (B-1)$$

$$13.78\dot{v} + 1.48u + 4.46v - 13.78\dot{w} = 0 \qquad (B-2)$$

$$\ddot{w} - 0.2095u + 0.574v + 2.322\dot{w} = 0 \qquad (B-3)$$

$$u\ (max) = 23 \qquad (B-4)$$

$$v\ (max) = 6.6 \qquad (B-5)$$

$$w\ (max) = 8.5 \qquad (B-6)$$

$$\dot{w}\ (max) = 6.2 \qquad (B-7)$$

Then defining

$$\begin{aligned} x_1 &= u \\ x_2 &= v \\ x_3 &= w \\ x_4 &= -\dot{w} = -\dot{x}_3 \end{aligned} \qquad (B-8)$$

The problem equations may now be converted to the standard form.

$$
\begin{aligned}
-13.78\dot{x}_1 &= 0.088x_1 + 0.392x_2 + 0.741x_3 \\
-13.78\dot{x}_2 &= 1.48x_1 + 4.46x_2 + 13.78x_4 \\
-\dot{x}_3 &= x_4 \\
-\dot{x}_4 &= 0.2095x_1 - 0.574x_2 + 2.322x_4
\end{aligned}
\qquad \text{(B-9)}
$$

Then in the matrix form ($-\dot{X}=AX$) the system becomes:

$$
-\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0.00635 & 0.0285 & 0.0538 & 0 \\ 0.1075 & 0.324 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.2095 & -0.574 & 0 & 2.322 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad \text{(B-10)}
$$

In applying each phase of the automatic program to the given problem which is now in the required form, a great deal of internal bookkeeping information is generated in arriving at the pertinent results which are of interest. Therefore, for the sake of clarity, only the specific results which are the outputs of the major phases of the automatic program are summarized for reference. A sample of all the printed statements resulting from the program are shown in Figure (B-1). Finally, the analog computer circuit diagram which results from carrying out the wiring directions is shown in Figure (B-2), and the circuit diagram for a manual solution is illustrated in Figure (B-3). The method of applying the scale factors calculated by the program to the scaling of problem variables and initial conditions is indicated on the computer circuit.

Results of Major Routines of the Automatic Program

Results of Problem Check Program

(1) AUM (1) = 3
AUM (2) = 3
AUM (3) = 1
AUM (4) = 3

(2) TAUM = 10

(3) AMAX = 2.332
AMIN = 0.00638

Results of Amplitude Scaling Program

$Z_1 = 4.35$ $S_1 = 4$

$Z_2 = 15.12$ $S_2 = 10$

$Z_3 = 11.18$ $S_3 = 10$

$Z_4 = 16.11$ $S_4 = 10$

$$E = \begin{bmatrix} 0.00638 & 0.01140 & 0.0538 & 0 \\ 0.26875 & 0.3240 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.52375 & -0.5740 & 0 & 2.322 \end{bmatrix}$$

Results of Time Scaling Program

$$G = \begin{bmatrix} 3 & 2 & 1 & 15 & 15 & 15 & 15 \\ 4 & 2 & 1 & 15 & 15 & 15 & 15 \\ 4 & 15 & 15 & 15 & 15 & 15 & 15 \\ 4 & 2 & 1 & 15 & 15 & 15 & 15 \end{bmatrix}$$

$T_1 = 4.30$

$T = 4.0$

$$E = \begin{bmatrix} 0.02552 & 0.04560 & 0.08608 & 0 \\ 1.07500 & 1.296 & 0 & 4 \\ 0 & 0 & 0 & 4 \\ 2.095 & -2.296 & 0 & 9.288 \end{bmatrix}$$

Results of Integrator Loading Alleviation and Amplifier Minimization Program

OKP (1)	3	OKP (2)	2	OKP (3)	1	OKP (4)	3
OKN (1)	0	OKN (2)	1	OKN (3)	0	OKN (4)	0
OK (1)	3	OK (2)	3	OK (3)	1	OK (4)	3

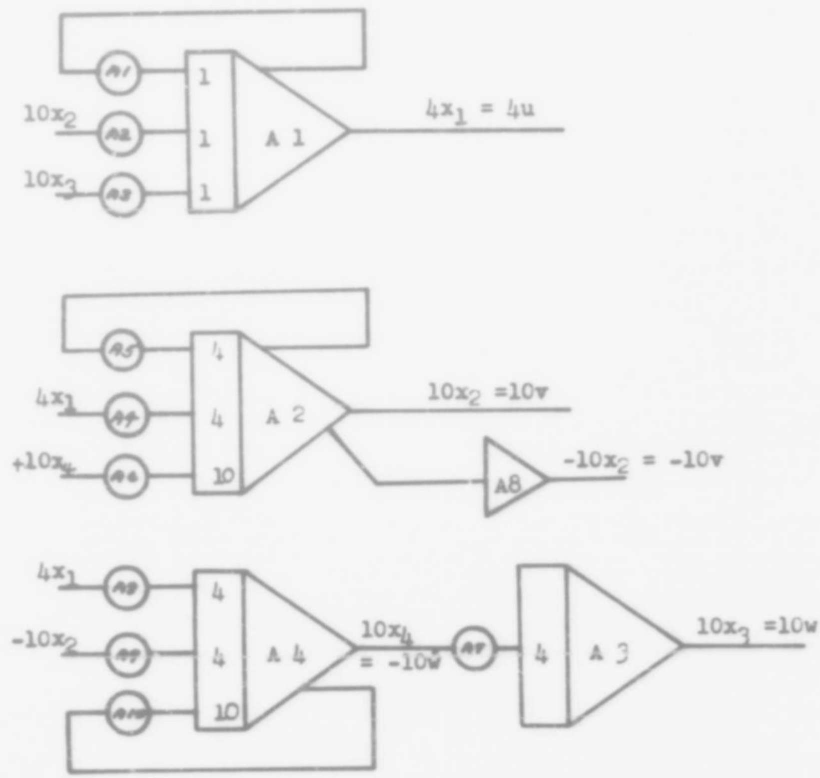
NEGI (1)	0	NEGI (2)	8	NEGI (3)	0	NEGI (4)	0
NOINT (1)	4	NOINT (2)	3	NOINT (3)	4	NOINT (4)	4
NONEG (1)	0	NONEG (2)	4	NONEG (3)	0	NONEG (4)	0
NEG2 (1)	0	NEG2 (2)	0	NEG2 (3)	0	NEG2 (4)	0

```
WIRING DIAGRAM
INT    A,  2   OUT
AMP    A,  8    1

INT    A,  1   OUT
POT    A,  1   IN

INT    A,  2   OUT
POT    A,  2   IN

INT    A,  3   OUT
POT    A,  3   IN

INT    A,  1   OUT
POT    A,  4   IN

INT    A,  2   OUT
POT    A,  5   IN

INT    A,  4   OUT
POT    A,  6   IN

INT    A,  4   OUT
POT    A,  7   IN

INT    A,  1   OUT
POT    A,  8   IN

AMP    A,  8   OUT
POT    A,  9   IN

INT    A,  4   OUT
POT    A, 10   IN
POT    A,  1   OUT
INT    A,  1        1.000E 00

POT    A,  2   OUT
INT    A,  1        1.000E 00

POT    A,  3   OUT
INT    A,  1        1.000E 00

POT    A,  4   OUT
INT    A,  2        4.000E 00

POT    A,  5   OUT
INT    A,  2        4.000E 00

POT    A,  6   OUT
INT    A,  2        1.000E 01
```

Figure B-1 (Part I)
Sample Problem Output

```
POT    A,  7    OUT
INT    A,  3         4.000E 00

POT    A,  8    OUT
INT    A,  4         4.000E 00

POT    A,  9    OUT
INT    A,  4         4.000E 00

POT    A, 10    OUT
INT    A,  4         1.000E 01


 POT SETTINGS
POT    A,  1         2.552E-02
POT    A,  2         4.560E-02
POT    A,  3         8.608E-02
POT    A,  4         2.687E-01
POT    A,  5         3.240E-01
POT    A,  6         4.000E-01
POT    A,  7        10.000E-01
POT    A,  8         5.237E-01
POT    A,  9         5.740E-01
POT    A, 10         9.288E-01


 X SCALE FACTORS
SCALE X 1    BY      4.000E 00
SCALE X 2    BY     10.000E 00
SCALE X 3    BY     10.000E 00
SCALE X 4    BY     10.000E 00


 POT NUMBER    RELATED COEFFICIENT
POT A,  1        1       1
POT A,  2        1       2
POT A,  3        1       3
POT A,  4        2       1
POT A,  5        2       2
POT A,  6        2       4
POT A,  7        3       4
POT A,  8        4       1
POT A,  9        4       2
POT A, 10        4       4
TIME SCALE FACTOR=    4.000E 00
```

Figure B-1 (Part II)
Sample Problem Output

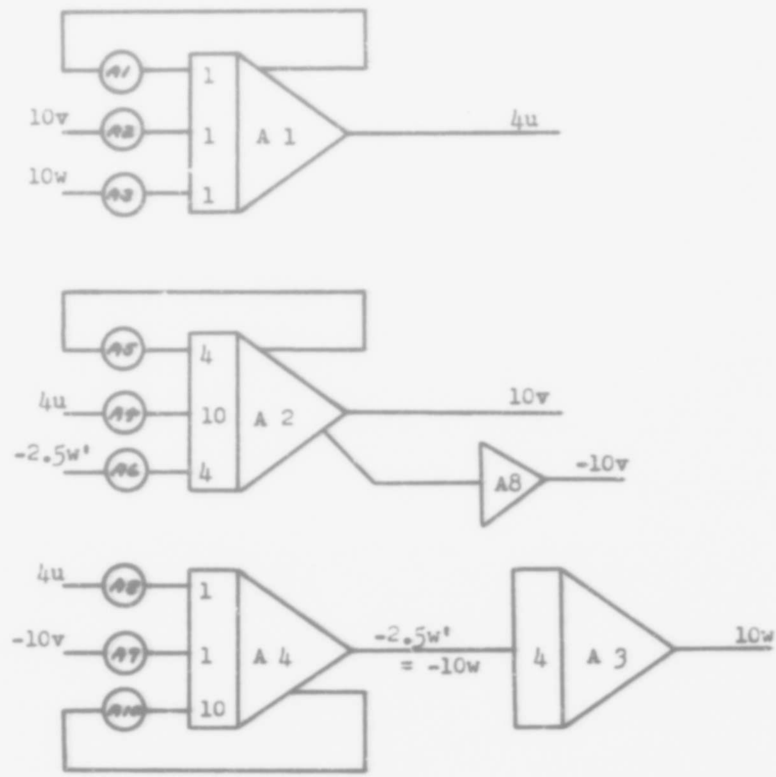


Pot	Settings
A-1	0.02552
A-2	0.04560
A-3	0.08608
A-4	0.2687
A-5	0.3240
A-6	0.4000
A-7	1.0000
A-8	0.5237
A-9	0.5740
A-10	0.9288

Time Scale Factor
4.0

Figure B-2

Analog Computer Circuit (Automatic)



Pot	Setting	Time Scale Factor
A-1	0.02552	4.0
A-2	0.04560	
A-3	0.08608	
A-4	0.269	
A-5	0.1296	
A-6	(not used)	
A-7	(not used)	
A-8	0.5237	
A-9	0.5740	
A-10	0.9288	

Figure B-3

Analog Computer Circuit (Manual)

Appendix C

Some Requirements of a More General Program

During the course of this investigation, several possibilities were explored for expanding the program so as to remove some of the restrictions of the present system. The basic principles and techniques established in Chapter II are applicable, for the most part, to such an extended program. However, for some of the higher-order systems within this category it is necessary to prescribe summing amplifiers and inverters to form the required integrator inputs, thereby considerably increasing the complexity of the program, and emphasizing the need for economy in the use of available equipment. This requirement has already been discussed at length in Chapter II. Furthermore, as mentioned in Chapter IV, the program illustrated in Appendix A will not, in general, yield as economical a solution as an analog operator might contrive. As a step toward an optimum solution, then, it is desirable to minimize the number of amplifiers required for the solution of a given problem.

Amplifier Minimization

The proposed problem, then, is to develop a technique for minimizing the number of amplifiers required in the solution of a given problem, subject to the restriction that loading effects at the integrator inputs and outputs are non-existent.

Preliminary Considerations. As has been stated previously, the rows of a given matrix indicate inputs to an integrator whereas the columns specify outputs from a given integrator. It follows that the implementation of positive elements (a_{ij}) requires no sign inversion between the output of integrator j and the input of integrator i. However, for negative elements (a_{ij}), the needed sign inversion between the output of integrator j and the input of integrator i must be obtained by (1) an inverting amplifier associated with the output of integrator j, or (2) an inverting amplifier associated with the input of integrator i. The two possibilities are illustrated in (a) and (b) below.

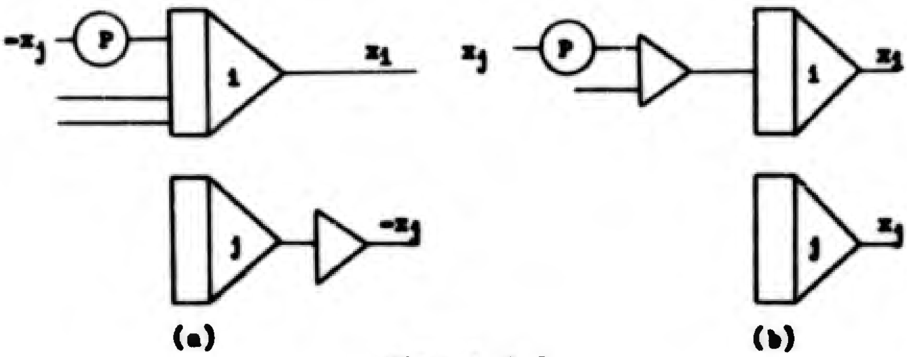


Figure C-1
(a) Amplifier at Integrator Output
(b) Amplifier at Integrator Input

This suggests two avenues of approach to the problem. Any technique by which one could maximize the number of columns containing all non-negative elements would reduce to a minimum the number of amplifiers required at integrator outputs. Further, any technique which would maximize the number of rows containing all non-negative elements would likewise reduce the number of amplifiers needed at integrator inputs.

Given a system of matrix equations,

$$-\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}a_{11} & -a_{12} & a_{13}\\ a_{21} & a_{22} & -a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}$$

Expanding these equations, we obtain

$$-\dot{x}_1 = a_{11}x_1 - a_{12}x_2 + a_{13}x_3$$
$$-\dot{x}_2 = a_{21}x_1 + a_{22}x_2 - a_{23}x_3$$
$$-\dot{x}_3 = a_{31}x_1 + a_{32}x_2 + a_{33}x_3$$

Then multiplying the second equation by -1

$$-\dot{x}_1 = a_{11}x_1 - a_{12}x_2 + a_{13}x_3$$
$$+\dot{x}_2 = -a_{21}x_1 - a_{22}x_2 + a_{23}x_3$$
$$-\dot{x}_3 = a_{31}x_1 + a_{32}x_2 + a_{33}x_3$$

and again arranging the equations in matrix form

$$-\begin{bmatrix}\dot{x}_1\\ -\dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}a_{11} & a_{12} & a_{13}\\ -a_{21} & a_{22} & a_{23}\\ a_{31} & -a_{32} & a_{33}\end{bmatrix}\begin{bmatrix}x_1\\ -x_2\\ x_3\end{bmatrix}$$

Thus, it has been illustrated that a change in sign in one equation of a system will cause a sign change in that row and associated column of the coefficient matrix. In other words, the multiplication of a row and its associated column of the coefficient matrix implies that to maintain the equality the signs of the variables corresponding to that row must be changed.

Figure C-2
Operation on Coefficient Matrix

Matrix Properties. A method which applies these ideas will be suggested after some additional properties of the matrix have been discussed.

1. The lower bound on the number of amplifiers needed corresponds to the number of negative elements along the principal diagonal of the matrix. Each such element is part of a feedback loop around its associated integrator. Consequently, each negative element along the diagonal generates a requirement for an inverting amplifier. Such columns may then be disregarded in the analysis, since operation on them will not decrease the number of amplifiers needed.

2. The upper bound on the number of amplifiers required will be equal to the order of the matrix.

3. Since only the signs of the elements of the coefficient matrix are pertinent in determining the need for amplifiers, it is convenient to replace the elements of the matrix with their respective signs. The elements of the resulting 'matrix of signs' may be changed at will by multiplying any row and its corresponding column by the factor (-1). Such manipulation of the matrix will result only in a change of sign of the problem variable associated with that row and column whose signs were changed. This is illustrated in Fig. (C-2), facing.

4. Columns which are 'like' -- in the sense that when considered side by side, adjacent elements are either identical throughout (+,+ or -,-) or opposite throughout (+,- or -,+) -- can be transformed into columns containing only non-negative elements by appropriate multiplication by the factor (-1). It may be noted that this statement does not apply to columns

containing a negative major diagonal term. In this case the particular column containing the negative major diagonal element cannot be made non-negative. It may also be pointed out that zero elements may be considered as either positive or negative in the determination of 'like' columns, but cannot be considered as both positive and negative in the determination of the total number of 'like' columns.

To illustrate the procedures discussed above, consider the matrix of signs shown below.

$$\begin{array}{ccc} (1) & (2) & (3) \end{array}$$
$$\begin{bmatrix} + & - & + \\ - & + & + \\ + & - & + \end{bmatrix}$$

Columns (1) and (2) are alike in the sense that adjacent elements are opposite in sign throughout. Multiplication of column (2) and its corresponding row by (-1) results in a matrix in which columns (1) and (2) contain only non-negative elements.

$$\begin{array}{ccc} (1) & (2) & (3) \end{array}$$
$$\begin{bmatrix} + & + & + \\ + & + & - \\ + & + & + \end{bmatrix}$$

Similarly, in a matrix in which columns are alike in the sense that adjacent elements are identical throughout -- (1) and (2) below --,

$$\begin{matrix} (1) & (2) & (3) \end{matrix}$$
$$\begin{bmatrix} + & + & + \\ + & + & + \\ - & - & + \end{bmatrix}$$

multiplication of the third row and its corresponding column by (-1) results in an equivalent matrix in which columns (1) and (2) contain only non-negative elements.

$$\begin{matrix} (1) & (2) & (3) \end{matrix}$$
$$\begin{bmatrix} + & + & - \\ + & + & - \\ + & + & + \end{bmatrix}$$

Similar arguments can be made for determining the 'likeness' of rows of a 'matrix of signs', and the same procedure can be applied to form rows which contain only non-negative elements.

5. Columns originally containing all positive elements must be considered, since any operations on the matrix will change the signs of elements in these columns.

6. The number of 'like' columns is invariant in the sense that they will remain 'alike' regardless of the operations performed on the matrix.

7. Elements with zero value may be disregarded in determining the 'likeness' of pairs of columns; that is, if the two columns are being compared, element by element -- only those elements adjacent to each other, where neither one is zero, need be considered.

8. Letting C_+ represent the number of columns having all positive elements, and P_+ the number of non-negative principal diagonal elements, then if $C_+ > P_+/2$, nothing can be gained by operating on the matrix. This can be verified by considering the following:

a. Any operation on the matrix will introduce negative elements into the C_+ columns.

b. Since the number of C_+ columns is greater than one-half the total number of columns which can, under any circumstances, be made positive, any non-trivial operation will result in fewer than C_+ all positive columns.

Minimization Techniques. On the basis of these preliminary observations and matrix properties, a technique for minimizing the number of inverting amplifiers required is as follows:

1. After determining that the inequality $C_+ > P_+/2$ is not satisfied, determine the columns of the matrix which are alike in either sense. The number of columns which are alike represents the maximum number of columns containing only non-negative elements which can be formed.

2. Of the columns selected, choose the one containing the least number of negative elements, and multiply each of the rows containing these negative elements (and their associated columns) by (-1). The elements of the resulting matrix indicate the signs which must be affixed to corresponding elements of the original coefficient matrix in order to require the minimum number of amplifiers at integrator outputs.

3. The same arguments can be applied to the rows of the coefficient matrix by interchanging the word 'row'

for the word 'column' in steps (1) and (2) above. The result will indicate the minimum number of inverting amplifiers required at the integrator inputs.

4. Finally, the decision to assign amplifiers at integrator outputs or inputs will depend upon which procedure results in a lesser number. That is, if the largest set of 'like' columns of step (2) is greater than the largest set of 'like' rows of step (3), then it will be more economical to assign amplifiers at the integrator outputs, and conversely.

Determination of 'Like' Columns. A rapid and convenient technique for determining the likeness of pairs of columns or rows involves pre-multiplication and post-multiplication of the original matrix by its transpose. The technique requires that the coefficient matrix of the system contain only non-negative elements along the principal diagonal. Its application is best illustrated by an example.

Consider the result of pre-multiplication of the given matrix a by its transpose A(TR).

$$\underset{A(TR)}{\begin{bmatrix} + & + & - & + \\ o & + & + & - \\ - & - & + & o \\ + & - & o & + \end{bmatrix}} \underset{A}{\begin{bmatrix} + & o & - & + \\ + & + & - & - \\ - & + & + & o \\ + & - & o & + \end{bmatrix}}$$

A product matrix is formed in which the individual elements are composed of n terms of two factors. Retaining only the signs resulting from the multiplication of the factors in each element position (without performing the indicated addition or subtraction), the matrix appears as shown below.

$$\begin{bmatrix} (++++) & (0+--) & (---0) & (+-0+) \\ (0+--) & (0+++) & (0-+0) & (0-0-) \\ (---0) & (0-+0) & (+++0) & (-+00) \\ (+-0+) & (0-0-) & (-+00) & (++0+) \end{bmatrix}$$

At this point it is advantageous to rewrite the matrix shown above in another form by defining two new operations as follows:

1. Let any element position containing either all positive or all negative signs (zeros ignored) be identically equal to unity.

2. Let any element position containing a combination of positive and negative signs (zeros ignored) be identically equal to zero.

Applying these definitions and rewriting, a new matrix is formed in which the subscripts of the element indicate the pairs of columns which are alike. That is, if $a_{ij}=1$, the columns represented by the numbers i and j are alike. Conversely if $a_{ij}=0$, the columns indicated by the subscripts are not alike. The resulting matrix is shown below.

$$B = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Only the portion above, or below, the principal diagonal need be considered since either provides the same information. On the basis of the given information, $B_{13}=B_{24}=1$. Therefore columns (1) and (3) and columns (2) and (4) of the original matrix are alike in the senses described earlier.

The same technique may be applied to the pairs of rows which are alike by post-multiplying the given matrix by its transpose. The subscripts of the resulting matrix as defined above will then specify the pairs of rows which are alike in the senses defined.

It is pertinent to note that in any case, it is necessary to determine, from the pairs of 'like' columns, the total number of 'like' columns. This total then represents the maximum number of integrators which will not require amplifiers.

Appendix D

The Automatic Program in Fortran

```
*61-14  922A  BAUMAN KEN    ANALOGUE
*      XEQ
*      LIST
CANALOG
       DIMENSION A(14,14),AUM(14),X(14),VXP1(46),LXP2(46),XP3(46),
      1IXP3(46),KYP4(46)
       DIMENSIONHO[illegible](1600),HOP1(500)
  500  READ INPUT TAPE 2,314,N
       READ INPUT TAPE 2,300,(X(I),I=1,N),((A(I,J),J=1,N),I=1,N)
       NUMI=0
       NUM2=0
       TAUM=0
       AMAX=0
       AMIN=0.1
    1  DO8I=1,N
       AUM(I)=0
    2  DO8J=1,N
       IF(ABSF(A(I,J)))8,8,3
    3  IF(ABSF(A(I,J))-AMAX)5,5,4
    4  AMAX=ABSF(A(I,J))
    5  IF(ABSF(A(I,J))-AM(N)6,6,7
    6  AMIN=ABSF(A(I,J))
    7  AUM(I)=AUM(I)+1.0
    8  CONTINUE
    9  DO12I=1,N
       IF(AUM(I)-7.0)11,11,10
   10  WRITE OUTPUT TAPE 3,301,I
       GOTO500
   11  TAUM=TAUM+AUM(I)
   12  CONTINUE
       IF(TAUM-46.0)14,14,13
   13  WRITE OUTPUT TAPE 3,302
       GOTO500
   14  CALL DUMP1(14,N,TAUM,A,AUM,X)
       DIMENSIONS(14)
       DO25I=1,N
       Z=100.0/ABSF(X(I))
       C=0
   15  D=10.0**C
       IF(D-Z)17,16,17
   16  B=1.0
       GO TO 24
   17  IF(D-Z)19,19,18
   18  C=C-1.0
       GO TO 15
```

```
19    IF(D-Z/10.0)20,20,21
20    C=C+1.0
      GO TO 15
21    B=9.0
22    IF(B*D-Z)24,24,23
23    B=B-1.0
      GO TO 22
24    S(I)=B*D
25    CONTINUE
      DIMENSION E(14,15)
26    DO27I=1,N
      DO27J=1,N
27    E(I,J)=A(I,J)*S(I)/S(J)
28    CALLDUMP2(28,N,E,S)
      DIMENSIONIG(14,7)
      DO29I=1,N
      DO29J=1,7
29    IG(I,J)=15
      DO30I=1,N
30    E(I,15)=0
31    DO39I=1,N
      DO39J=1,N
      IF(ABSF(E(I,J)))39,39,32
32    L=1
33    K=IG(I,L)
      IF(ABSF(E(I,J))-ABSF(E(I,K)))34,34,35
34    L=L+1
      GO TO 33
35    M=7
36    MO=M-1
      IG(I,M)=IG(I,MO)
      IF(M-(L+1))38,38,37
37    M=M-1
      GO TO 36
38    M=M-1
      IG(I,M)=J
39    CONTINUE
48    CALL DUMP 4(48,N,AMAX,AMIN,IG)
      IF(AMAX-10.0)40,40,42
40    IF(AMIN-0.1)42,42,41
41    T1=1.0
      GO TO 43
42    T1=10.0/AMAX
43    DO47I=1,N
      K=IG(I,3)
```

```
      IF(ABSF(E(I,K))-4.0/T1)45,45,44
44    T1=4.0/ABSF(E(I,K))
45    K=IG(I,5)
      IF(ABSF(E(I,K))-1.0/T1)47,47,46
46    T1=1.0/ABSF(E(I,K))
47    CONTINUE
      C=0
49    D=10.0**C
      IF(D-T1)51,50,51
50    B=1.0
      GO TO 58
51    IF(D-T1)53,53,52
52    C=C-1.0
      GO TO 49
53    IF(D-T1/10.0)54,54,55
54    C=C+1.0
      GO TO 49
55    B=9.0
56    IF(B*D-T1)58,58,57
57    B=B-1.0
      GO TO 56
58    I=B*D
      DO59I=1,N
      DO59J=1,N
59    E(I,J)=E(I,J)*T
60    CALLDUMP2(60,N,E,S)
      DIMENSION NOKP(14),NOKN(14),NOK(14)
61    DO66J=1,N
      NOKP(J)=0
      NOKN(J)=0
      NOK(J)=0
62    DO66I=1,N
      IF(E(I,J))104,66,64
104   NOKN(J)=NOKN(J)+1
      GO TO 65
64    NOKP(J)=NOKP(J)+1
65    NOK(J)=NOK(J)+1
66    CONTINUE
67    CALLDUMP3(67,N,NOKP,NOKN,NOK)
      NPRIN=0
      NPRI=0
      NPT=0
      NPRINT=0
      NSUMA=7
      NSUMB=7
```

```
      NINVA=14
      NINVB=14
      NINR=100
      NPOTA=0
      NPOTB=0
      DIMENSION NEG1(14),NEG2(14),NPOS(14),NOINT(14),
     1NONEG(14)
      DO63J=1,N
      NEG1(J)=0
      NEG2(J)=0
      NPOS(J)=0
      NOINT(J)=0
      NONEG(J)=0
      IF(NOK(J)-4)68,68,70
   68 IF(NOKN(J))69,69,78
   69 NOINT(J)=4
      GO TO 63
   70 IF(NOK(J)-7)71,71,73
   71 IF(NOKP(J)-3)72,72,79
   72 IF(NOKN(J)-4)78,78,81
   73 IF(NOKN(J)-3)74,74,75
   74 IF(NOKP(J)-7)79,79,77
   75 IF(NOKP(J)-6)76,76,77
   76 IF(NOKN(J)-7)80,80,77
   77 WRITE OUTPUT TAPE 3,305
      GOTO500
   78 LEAVE=0
      GO TO 82
   79 LEAVE=1
      GO TO 82
   80 LEAVE =2
      GO TO 82
   81 LEAVE=3
   82 NPRINT=NPRINT+1
      IF(J-7)83,83,94
   83 NSUMA=NSUMA+1
      IF(NSUMA-14)84,84,93
   84 CALL PRINT(1,J,NSUMA)
      NEG1(J)=NSUMA
      IF(LEAVE)89,89,85
   85 IF(LEAVE-2)86,86,120
   86 NPRINT=NPRINT+1
      NONEG(J)=3
      NINVA=NINVA+1
      IF(NINVA-20)87,87,90
```

```
87    CALL PRINT(3,NSUMA,NINVA)
      NPOS(J)=NINVA
92    IF(LEAVE-1)88,88,120
88    NOINT(J)=3
      GO TO 63
89    NOINT(J)=3
      NONEG(J)=4
      GO TO 63
90    NSUMA=NSUMA+1
      IF(NSUMA-14)91,91,93
91    CALL PRINT(3,NSUMA-1,NSUMA)
      NPOS(J)=NSUMA
      GO TO 92
93    WRITE OUTPUT TAPE 3,306
      GOTO500
94    NINT=J-7
      NSUMB=NSUMB+1
      IF(NSUMB-14)105,105,93
105   CALL PRINT(2,NINT,NSUMB)
      NEG1(J)=NSUMB
      IF(LEAVE)89,89,95
95    IF(LEAVE-2)124,124,123
124   NPRINT=NPRINT+1
      NONEG(J)=3
      NINVB=NINVB+1
      IF(NINVB-20)96,96,98
96    CALL PRINT(4,NSUMB,NINVB)
      NPOS(J)=NINVB
97    IF(LEAVE-1)88,88,123
98    NSUMB=NSUMB+1
      IF(NSUMB-14)99,99,93
99    CALL PRINT(4,NSUMB-1,NSUMB)
      NPOS(J)=NSUMB
      GO TO 97
120   NPRINT=NPRINT+1
      NINVA=NINVA+1
      IF(NINVA-20)121,121,122
121   CALL PRINT(1,J,NINVA)
      NEG2(J)=NINVA
126   NOINT(J)=2
63    CONTINUE
      GO TO 129
122   NSUMA=NSUMA+1
      IF(NSUMA-14)106,106,93
106   CALL PRINT(1,J,NSUMA)
```

```
      NEG2(J)=NSUMA
      GO TO 126
123   NPRINT=NPRINT+1
      NINVB=NINVB+1
      IF(NINVB-20)125,125,127
125   CALL PRINT(2,NINT,NINVB)
      NEG2(J)=NINVB
      GO TO 126
127   NSUMB=NSUMB+1
      IF(NSUMB-14)128,128,93
128   CALL PRINT(2,NINT,NSUMB)
      NEG2(J)=NSUMB
      GO TO 126
129   IF(TAUM-23.0)131,131,130
130   MOVE=1
      GO TO 132
131   MOVE=0
132   DO194I=1,N
      PSET=0
      NGN4=3
      NGN1=4
      GN=0
      L=7
133   IF(IG(I,L)-14)136,136,134
134   GO TO 192
136   K=IG(I,L)
      NPRINT=NPRINT+1
      IF(K-7)137,137,152
137   NPOTA=NPOTA+1
      IF(NPOTA-23)138,138,142
138   M=0
      IF(E(I,K))139,139,100
139   NONEG(K)=NONEG(K)-1
      IF(NONEG(K))140,140,141
140   CALL PRINT(6,NEG2(K),NPOTA)
      GO TO 171
141   CALL PRINT(6,NEG1(K),NPOTA)
      GO TO 171
100   NOINT(K)=NOINT(K)-1
      IF(NOINT(K))101,101,102
101   CALL PRINT(6,NPOS(K),NPOTA)
      GO TO 171
102   CALL PRINT(5,K,NPOTA)
      GO TO 171
142   M=1
```

```
      NPOTB=NPOTB+1
      NINR=NINR+1
      IF(NINR-122)145,145,143
143   NINR=NINR+78
      IF(NINR-222)145,145,144
144   WRITE OUTPUT TAPE 3,307
      GOTO500
145   IF(E(I,K))146,146,148
146   NONEG(K)=NONEG(K)+1
      IF(NONEG(K))103,103,147
147   CALL PRINT(14,NEG1(K),NINR)
      GO TO 151
103   CALL PRINT(14,NEG2(K),NINR)
      GO TO 151
148   NOINT=NOINT+1
      IF(NOINT)150,150,149
149   CALL PRINT(7,K,NINR)
      GO TO 151
150   CALL PRINT(14,NPOS(K),NINR)
151   NPRINT=NPRINT+1
      CALL PRINT(8,NINR,NPOTB)
      GO TO 171
152   IF(MOVE)163,163,162
163   NPOTA=NPOTA+1
      K=0
      NINR=NINR+1
      IF(NINR-122)154,154,153
153   NINR=NINR+78
      IF(NINR-222)154,154,144
154   IF(E(I,K))155,155,158
155   NONEG(K)=NONEG(K)-1
      IF(NONEG(K))156,156,157
156   CALL PRINT(10,NEG2(K),NINR)
      GO TO 161
157   CALL PRINT(10,NEG1(K),NINR)
      GO TO 161
158   NOINT(K)=NOINT(K)-1
      IF(NOINT(K))159,159,160
159   CALL PRINT(10,NPOS(K),NINR)
      GO TO 161
160   CALL PRINT(9,K-7,NINR)
161   NPRINT=NPRINT+1
      CALL PRINT(11,NINR,NPOTA)
      GO TO 171
162   NPOTB=NPOTB+1
```

```
      IF(NPOTB-23)164,164,163
164   M=1
      IF(E(I,K))165,165,168
165   NONEG(K)=NONEG(K)+1
      IF(NONEG(K))167,167,166
166   CALL PRINT(13,NEG1(K),NPOTB)
      GO TO 171
167   CALL PRINT(13,NEG2(K),NPOTB)
      GO TO 171
168   NOINT(K)=NOINT(K)-1
      IF(NOINT(K))169,169,170
169   CALL PRINT(13,NPOS(K),NPOTB)
      GO TO 171
170   CALL PRINT(12,K-7,NPOTB)
171   IF(ABSF(E(I,K))-1.001)176,176,172
172   IF(ABSF(E(I,K))-4.001)173,173,174
173   NGN4=NGN4-1
      IF(NGN4)174,174,175
174   GN=10.0
      GO TO 178
175   GN=4.0
      GO TO 178
176   NGN1=NGN1-1
      IF(NGN1)173,173,177
177   GN=1.0
178   NPRIN=NPRIN+1
      PSET=ABSF(E(I,K))/GN
      XP3(NPRIN)=PSET
      NPRINT=NPRINT+1
      NPT=NPT+1
      IXP3(NPT)=I
      KXP4(NPT)=K
      IF(M)179,179,180
179   LXP2(NPT)=NPOTA
B     VXP1(NPT)=474663602174
      GO TO 181
180   LXP2(NPT)=NPOTB
B     VXP1(NPT)=474663602274
181   IF(M)187,187,182
182   IF(I-7)184,184,183
183   CALL PRINT(17,NPOTB,I-7,GN)
      GOTO192
184   NINR=NINR+1
      IF(NINR-122)186,186,185
185   NINR=NINR+78
```

```
      IF(NINR-222)186,186,144
186   CALL PRINT(15,NPOTB,NINR)
      NPRINT=NPRINT+1
      CALL PRINT(18,NINR,I,GN)
      GOTO192
187   IF(I-7)188,188,189
188   CALL PRINT(19,NPOTA,I,GN)
      GO TO 192
189   NINR=NINR+1
      IF(NINR-122)191,191,190
190   NINR=NINR+78
      IF(NINR-122)191,191,144
191   CALLPRINT(16,NPOTA,NINR)
      NPRINT=NPRINT+1
      CALL PRINT(20,NINR,I-7,GN)
192   IF(L-1)194,194,193
193   L=L-1
      GO TO 133
194   CONTINUE
195   IP=8.0*TAUM
      J=8*NPRINT-IP
400   WRITE OUTPUT TAPE 3,308
401   WRITE OUTPUT TAPE3,309,(HOP(I),I=1,J)
409   WRITE OUTPUT TAPE3,318,(HOP1(I),I=1,IP)
406   WRITE OUTPUT TAPE3,315
402   WRITE OUTPUT TAPE 3,310,(VXP1(I),LXP2(I),XP3(I),I=1,NPRIN)
407   WRITE OUTPUT TAPE3,316
      DO403I=1,N
403   WRITE OUTPUT TAPE3,311,I,S(I)
408   WRITE OUTPUT TAPE3,317
404   WRITE OUTPUT TAPE 3,312,(VXP1(I),LXP2(I),IXP3(I),KXP4(I),I=1,NPT)
405   WRITE OUTPUT TAPE 3,313,T
      GO TO 500
300   FORMAT(6F12.0)
301   FORMAT(1H1,10X,33HTOO MANY INPUTS TO INTEGRATOR NO=I2)
302   FORMAT(1H1,10X,22HTOO MANY POTS REQUIRED)
305   FORMAT(1H1,10X,16HTOO MANY OUTPUTS)
306   FORMAT(1H1,10X,18HTOO MANY AMPLIFERS)
307   FORMAT(1H1,10X,25HTOO MANY INTERCONNECTIONS)
308   FORMAT(1H1,5X,14HWIRING DIAGRAM)
309   FORMAT(2A6,I3,A6/2A6,I3,A6/)
310   FORMAT(4X,1A6,I3,3X,1P1E12.3)
311   FORMAT(11H    SCALE X,I2,3X,2HBY,2X,1P1E12.3)
312   FORMAT(4X,1A6,I3,3X,I3,3X,I3)
313   FORMAT(22H    TIME SCALE FACTOR=1P1E12.3)
```

```
 314  FORMAT(I2)
 315  FORMAT(///5X,12HPOT SETTINGS)
 316  FORMAT(///5X,15HX SCALE FACTORS)
 317  FORMAT(///5X,32HPOT NUMBER   RELATED COEFFICIENT)
 318  FORMAT(2A6,I3,A6/2A6,I3,3X,1P1E12.3/)
      COMMONHOP,HOP1,NUM1,NUM2
      END
*     LIST
      SUBROUTINE DUMP1(NO,N,TAUM,A,AUM,X)
      DIMENSION A(14,14),AUM(14),X(14)
      WRITE OUTPUT TAPE 3,303,NO,N,TAUM,((A(I,J),J=1,N),I=1,N),
     1(X(I),I=1,N)
      RETURN
 303  FORMAT(1H1,I2,10X,I2,2X,1P1E12.4,6(2X,1P1E12.4)/(8(E12.4)))
      END
*     LIST
      SUBROUTINEDUMP2(NO,N,E,S)
      DIMENSION E(14,15),S(14)
      WRITE OUTPUT TAPE3,304,NO,((E(I,J),J=1,N),I=1,N),
     1(S(I),I=1,N)
      RETURN
 304  FORMAT(1X,I2,10X,7(2X,1P1E12.4)/(8(E12.4)))
      END
      SUBROUTINEDUMP3(NO,N,NOKP,NOKN,NOK)
      DIMENSION NOKP(14),NOKN(14),NOK(14)
      WRITE OUTPUT TAPE3,304,NO,(NOK(I),I=1,N),(NOKN(I),I=1,N)
     1,(NOKP(I),I=1,N)
      RETURN
 304  FORMAT(1X,I2,10X,7(2X,I5)/(8(2X,I5)))
      END
      SUBROUTINE DUMP 4(NO,N,AMAX,AMIN,IG)
      DIMENSION IG(14,14)
      WRITE OUTPUT TAPE 3,305,NO,AMAX,AMIN,((IG(I,J),J=1,N),I=1,N)
      RETURN
 305  FORMAT(1X,I2,10X,2(1P1E12.4)/(8(2X,I5)))
      END
*     SUBROUTINE PRINT STORE
*     LIST
*     FAP
       ENTRY     PRINT
PRINT  SXD       SAVE,4               SAVE
       SXD       SAVE+1,2             I
       SXD       SAVE+2,1             RS
       CLA       =0000001000000
       STO       CNT
```

```
         CLA    =0
         PAC    ,1
M1       CLA*   1,4                 PICK
         SUB    CNT                 PROPER
         TZE    STORE,1             FORMAT
         CLA    CNT                 TO
         ADD    =0000001000000
         STO    CNT
         TXI    M1,1,-13
STORE    CLA    T1
         STO    B1
         CLA    T1+1
         STO    B2
         CLA    T2
         STO    B3
         CLA    T5
         STO    B4
         CLA    T5+1
         STO    B5
         CLA    T6
         STO    B6
         TRA    BIN
NO2      CLA    T3
         STO    B1
         CLA    T3+1
         STO    B2
         CLA    T2
         STO    B3
         CLA    T7
         STO    B4
         CLA    T7+1
         STO    B5
         CLA    T6
         STO    B6
         TRA    BIN
NO3      CLA    T5
         STO    B1
         CLA    T5+1
         STO    B2
         CLA    T2
         STO    B3
         CLA    T5
         STO    B4
         CLA    T5+1
         STO    B5
```

```
      CLA   T6      497
      STO   B6      498
      TRA   BIN     499
NO4   CLA   T7      50
      STO   B1      501
      CLA   T7+1    502
      STO   B2      503
      CLA   T2      504
      STO   B3      505
      CLA   T7      506
      STO   B4      507
      CLA   T7+1    508
      STO   B5      509
      CLA   T6      51
      STO   B6      511
      TRA   BIN     512
NO5   CLA   T1      513
      STO   B1      514
      CLA   T1+1    515
      STO   B2      516
      CLA   T2      517
      STO   B3      518
      CLA   T8      519
      STO   B4      52
      CLA   T8+1    521
      STO   B5      522
      CLA   T9      523
      STO   B6      524
      TRA   BIN     525
NO6   CLA   T5      526
      STO   B1      527
      CLA   T5+1    528
      STO   B2      529
      CLA   T2      53
      STO   B3      531
      CLA   T8      532
      STO   B4      533
      CLA   T8+1    534
      STO   B5      535
      CLA   T9      536
      STO   B6      537
      TRA   BIN     538
NO7   CLA   T1      539
      STO   B1      54
      CLA   T1+1    541
```

```
        STO     B2                                   542
        CLA     T2                                   543
        STO     B3                                   544
        CLA     T11                                  545
        STO     B4                                   546
        CLA     T11+1                                547
        STO     B5                                   548
        CLA     T4                                   549
        STO     B6                                   55
        TRA     BIN                                  551
NO8     CLA     T.1                                  552
        STO     B1                                   553
        CLA     T11+1                                554
        STO     B2                                   555
        CLA     T4                                   556
        STO     B3                                   557
        CLA     T10                                  558
        STO     B4                                   559
        CLA     T10+1                                56
        STO     B5                                   561
        CLA     T9                                   562
        STO     B6                                   563
        TRA     BIN                                  564
NO9     CLA     T3                                   565
        STO     B1                                   566
        CLA     T3+1                                 567
        STO     B2                                   568
        CLA     T2                                   569
        STO     B3                                   57
        CLA     T11                                  571
        STO     B4                                   572
        CLA     T11+1                                573
        STO     B5                                   574
        CLA     T4                                   575
        STO     B6                                   576
        TRA     BIN                                  577
NO10    CLA     T7                                   578
        STO     B1                                   579
        CLA     T7+1                                 58
        STO     B2                                   581
        CLA     T2                                   582
        STO     B3                                   583
        CLA     T11                                  584
        STO     B4                                   585
        CLA     T11+1                                586
```

```
        STO    B5                                   587
        CLA    T4                                   588
        STO    B6                                   589
        TRA    BIN                                  59
NO11    CLA    T11                                  591
        STO    B1                                   592
        CLA    T11+1                                593
        STO    B2                                   594
        CLA    T4                                   595
        STO    B3                                   596
        CLA    T8                                   597
        STO    B4                                   598
        CLA    T8+1                                 599
        STO    B5                                   60
        CLA    T9                                   601
        STO    B6                                   602
        TRA    BIN                                  603
NO12    CLA    T3                                   604
        STO    B1                                   605
        CLA    T3+1                                 606
        STO    B2                                   607
        CLA    T2                                   608
        STO    B3                                   609
        CLA    T10                                  61
        STO    B4                                   611
        CLA    T10+1                                612
        STO    B5                                   613
        CLA    T9                                   614
        STO    B6                                   615
        TRA    BIN                                  616
NO13    CLA    T7                                   617
        STO    B1                                   618
        CLA    T7+1                                 619
        STO    B2                                   62
        CLA    T2                                   621
        STO    B3                                   622
        CLA    T10                                  623
        STO    B4                                   624
        CLA    T10+1                                625
        STO    B5                                   626
        CLA    T9                                   627
        STO    B6                                   628
        TRA    BIN                                  629
NO14    CLA    T5                                   63
        STO    B1                                   631
```

```
         CLA    T5+1                         632
         STO    B2                           633
         CLA    T2                           634
         STO    B3                           635
         CLA    T11                          636
         STO    B4                           637
         CLA    T11+1                        638
         STO    B5                           639
         CLA    T4                           64
         STO    B6                           641
         TRA    BIN                          642
NO15     CLA    T10                          643
         STO    B1                           644
         CLA    T10+1                        645
         STO    B2                           646
         CLA    T2                           647
         STO    B3                           648
         CLA    T11                          649
         STO    B4                           65
         CLA    T11+1                        651
         STO    B5                           652
         CLA    T4                           653
         STO    B6                           654
         TRA    BIN                          655
NO16     CLA    T8                           656
         STO    B1                           657
         CLA    T8+1                         658
         STO    B2                           659
         CLA    T4                           66
         STO    B3                           661
         CLA    T11                          662
         STO    B4                           663
         CLA    T11+1                        664
         STO    B5                           665
         CLA    T4                           666
         STO    B6                           667
         TRA    BIN                          668
NO17     CLA    T10                          669
         STO    B1                           67
         CLA    T10+1                        671
         STO    B2                           672
         CLA    T2                           673
         STO    B3                           674
         CLA    T3                           675
         STO    B4                           676
```

```
         CLA    T3+1                                 677
         STO    B5                                   678
         CLA*   4,4                                  679
         STO    B6                                   68
         TRA    BIN2                                 681
NO18     CLA    T11                                  682
         STO    B1                                   683
         CLA    T11+1                                684
         STO    B2                                   685
         CLA    T4                                   686
         STO    B3                                   687
         CLA    T1                                   688
         STO    B4                                   689
         CLA    T1+1                                 69
         STO    B5                                   691
         CLA*   4,4                                  692
         STO    B6                                   693
         TRA    BIN2                                 694
NO19     CLA    T8                                   695
         STO    B1                                   696
         CLA    T8+1                                 697
         STO    B2                                   698
         CLA    T2                                   699
         STO    B3                                   70
         CLA    T1                                   701
         STO    B4                                   702
         CLA    T1+1                                 703
         STO    B5                                   704
         CLA*   4,4                                  705
         STO    B6,                                  706
         TRA    BIN2                                 707
NO20     CLA    T11                                  708
         STO    B1                                   709
         CLA    T11+1                                71
         STO    B2                                   711
         CLA    T4                                   712
         STO    B3                                   713
         CLA    T3                                   714
         STO    B4                                   715
         CLA    T3+1                                 716
         STO    B5                                   717
         CLA*   4,4                                  718
         STO    B6                                   719
         TRA    BIN2                                 72
BIN      CLA    NUM1           STORE                 721
```

```
          PAX     ,2
          CLA     B1               IN
          STO     HOP,2
          CLA     B2               HOPPER
          STO     HOP-1,2
          CLA*    2,4
          STO     HOP-2,2
          CLA     B3
          STO     HOP-3,2
          CLA     B4
          STO     HOP-4,2
          CLA     B5
          STO     HOP-5,2
          CLA*    3,4
          STO     HOP-6,2
          CLA     B6
          STO     HOP-7,2
          CLA     NUM1
          ADD     =8
          STO     NUM1
          LXD     SAVE,4
          LXD     SAVE+1,2
          LXD     SAVE+2,1
          TRA     4,4
BIN2      CLA     NUM2
          PAX     ,2
          CLA     B1
          STO     HOP1,2
          CLA     B2
          STO     HOP1-1,2
          CLA*    2,4
          STO     HOP1-2,2
          CLA     B3
          STO     HOP1-3,2
          CLA     B4
          STO     HOP1-4,2
          CLA     B5
          STO     HOP1-5,2
          CLA*    3,4
          STO     HOP1-6,2
          CLA     B6
          STO     HOP1-7,2
          CLA     NUM2
          ADD     =8
          STO     NUM2
```

```
         LXD     SAVE,4                          767
         LXD     SAVE+1,2                        768
         LXD     SAVE+2,1                        769
         TRA     5,4                             77
SAVE     HTR                                     771
         HTR                                     772
         HTR                                     773
T1       BCI     2,   INT   A,                   774
T2       BCI     1,  OUT                         775
T3       BCI     2,   INT   B,                   776
T4       BCI     1,                              777
T5       BCI     2,   AMP   A,                   778
T6       BCI     1,   1                          779
T7       BCI     2,   AMP   B,                   78
T8       BCI     2,   POT   A,                   781
T9       BCI     1,  IN                          782
T10      BCI     2,   POT   B,                   783
T11      BCI     2,   INR                        784
B1       PZE                                     785
B2       PZE                                     786
B3       PZE                                     787
B4       PZE                                     788
B5       PZE                                     789
B6       PZE                                     79
CNT      PZE                                     791
HOP      COMMON  1600                            792
HOP1     COMMON  500                             793
NUM1     COMMON  1                               794
NUM2     COMMON  1                               795
         END                                     796
*       DATA                                     797
```

A . . . . Input coefficient matrix.

a_{ij} = A (I,J) . . General element of the coefficient matrix.

AUM (I) . . . . Number of non-zero elements in the i-th row of the coefficient matrix.

TAUM . . . . . . Total number of non-zero elements in the coefficient matrix.

AMAX . . . . . . Maximum absolute value of the elements in the coefficient matrix.

AMIN . . . . . . Minimum absolute value of the (non-zero) elements of the coefficient matrix.

S_i = S(I) . . . . Optimum amplitude ~~scale~~ factor for the i-th row of the coefficient matrix.

S . . . . Diagonal matrix containing the amplitude scale factors.

E' = SAS⁻ . . . . . Amplitude scaled matrix.

E'_{ij}=E' (I,J) . . General element of the amplitude scaled matrix.

E = TSAS . . . . . Time and amplitude scaled matrix.

E_{ij} = E (I,J) . . General element of the time and amplitude scaled matrix.

NOTE: In the digital computer, both the amplitude scaled matrix and the time and amplitude scaled matrix are stored in the same memory locations (E) at different times.

G . . . . Matrix of subscripts which is formed by arranging the elements of the rows of the amplitude scaled matrix in decending order and recording their column subscripts.

G (I,L) . . . . General element of the G matrix which is equal the column (j) subscript of the element of the E matrix which is the L-th largest in the i-th row.

λ = T . . . . . Optimum time scale factor.

OKP (J) . . . . Number of elements of the j-th column of the coefficient matrix which are greater than zero.

OKN (J) . . . . Number of elements of the j-th column of the coefficient matrix which are less than zero.

1

of the
efficient
the
lements of
row of the
cale factors.
trix.
scaled matrix.
ed matrix
tored in the
ranging the
aled matrix
olumn
qual the
he E matrix
w.
he
zero.
he
ro.

OK (J) . . . . Total number of non-zero elements in the j-th column of the coefficient matrix.

NPRINT . . . . Count of PRINT storage locations.

NPRIN . . . . . Count of PRIN storage locations.

NPRI . . . . . Count of PRI storage locations.

NPT . . . . . . Count of PT storage locations.

PRINT (NPRINT) Wiring statement to be printed out.

PRIN (NPRIN) . Potentiometer setting to be printed out.

PRI (NPRI) . . Amplitude scale factor to be printed out.

PT (NPT) . . . Statement, relating potentiometer to element of coefficient matrix, to be printed out.

NSUMA . . . . . Number of summer on bay A of the computer.

NSUMB . . . . . Number of summer on bay B of the computer.

NINVA . . . . . Number of inverter on bay A of the computer.

NINVB . . . . . Number of inverter on bay B of the computer.

NPOTA . . . . . Number of potentiometer on bay A of the computer.

NPOTB . . . . . Number of potentiometer on bay B of the computer.

NINR . . . . . Number of interconnection.

NEG1 (J) . . . Number of first amplifier assigned to integrator number J.

NEG2 (J) . . . Number of second amplifier assigned to integrator number J.

NPOS (J) . . . Number of cascaded amplifier assigned to integrator number J.

NOINT (J) . . Number of open (useable) outputs on integrator number J.

NONEG (J) Number of open (useable) outputs on the first amplifier assigned to integrator number J.

Vita

James [redacted] Wine was born on [redacted] in [redacted], the son of Clarence Wine and Anna [redacted] Wine. After graduating from [redacted] in 1952, he enrolled in Technological College, [redacted]. He was graduated in May 1956 with the degree of Bachelor of Science in Electrical Engineering. After receiving his commission as Lieutenant in the USAF, he entered active duty in February 1957. He served as an electrical engineer assigned to the National Security Agency before entering the Air Force Institute of Technology for graduate study in September 1959.

Permanent address: [redacted]

This thesis was typed by Mrs. Marjorie Felton.

Vita

William J[redacted] Reisinger was born on [redacted], the son of Joseph and Mary Reisinger, in [redacted]. He attended public schools in [redacted], and graduated from [redacted]. He joined the Bethlehem Steel Company and served as an apprentice machinist until late in 1943 when he enlisted as a private in the Army Air Corps. He subsequently entered flight training, receiving his wings and a commission in early 1945. After a short tour as a navigator instructor he completed Flight Engineer Training, receiving a second set of wings late in 1945. Following a two year lapse in service to study at Lehigh University, Bethlehem, Pennsylvania, he returned to active duty to attend Electronics Officer School at Keesler Air Force Base, Mississippi. After graduation followed successive tours, in Alaska as a line navigator with the 54th Troop Carrier Squadron, and at Ellington Air Force Base, Texas, as an electronics instructor. He graduated from the Air Force Institute of Technology with a Bachelor of Science Degree in Electrical Engineering in 1954. Following a tour in Germany as commander of a Matador control and guidance detachment, he returned to the Air Force Institute of Technology for graduate study in September, 1959.

Permanent address: [redacted]

This thesis was typed by Mrs. Marjorie Felton.

UNCLASSIFIED

UNCLASSIFIED