2(mix)

NASA TECHNICAL NOTE

NASA TN D-7621

NASA TN D-7621

REDUCE II - A COMPUTER PROGRAM FOR THE SYMBOLIC REDUCTION OF LARGE BLOCK DIAGRAMS

by Carl F. Lorenzo and John P. Riehl

Lewis Research Center

Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MAY 1974

100

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TN D-7621		

4. Title and Subtitle	5. Report Date
REDUCE II - A COMPUTER PROGRAM FOR THE SYMBOLIC REDUCTION OF LARGE BLOCK DIAGRAMS	MAY 1974
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Carl F. Lorenzo and John P. Riehl	E-5860
	10. Work Unit No.
9. Performing Organization Name and Address	501-24
Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Note
National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

REDUCE II is a FORMAC program which symbolically calculates the transfer function(s) of any linear-block-diagram output variable to any or all input variables. The program requires as input a set of algebraic expressions representing the block diagram, the desired transfer function(s), and a string of variables indicating the desired order of reduction. The solution is presented in the compact form of a set of nested functions (super G's). The program can handle systems as large as 600 equations and is intended as a tool for the analysis of complex control and dynamic systems. A companion FORTRAN program, EVAL II, which numerically evaluates the solution set to obtain amplitude ratio and phase angle as functions of frequency is also presented.

17. Key Words (Suggested by Author(s))

FORMAC
Symbolic manipulation
Block diagram reduction
Block diagrams
Dynamics
Control

18. Distribution Statement

Unclassified - unlimited

CAT-08

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	98	$4.00

* For sale by the National Technical Information Service, Springfield, Virginia 22151

CONTENTS

REDUCE II - A COMPUTER PROGRAM FOR THE SYMBOLIC REDUCTION OF LARGE BLOCK DIAGRAMS

by Carl F. Lorenzo and John P. Riehl

Lewis Research Center

SUMMARY

A computer program, REDUCE II, has been written to reduce very large (linear) block diagrams. The program requires as input a set of algebraic expressions representing the block diagram, the desired transfer function(s), and a string of variables indicating the desired order of reduction. REDUCE II then calculates the transfer function(s) of any output variable to any or all input variables. The program is premised on reduction, by loops, of the block diagram. The solution is presented in the compact form of a set of nested functions (super G's). The super G's define the transfer functions in terms of the system information (G's) and the preceding super G's.

The program can handle as many as 600 equations and 600 variables and is intended as a tool for the analysis of complex control and dynamic systems. Several applications are presented.

A FORTRAN evaluation program, EVAL II, accepts the symbolic output generated by REDUCE II and evaluates it numerically to obtain amplitude ratio and phase angle as functions of frequency for any desired G functions.

INTRODUCTION

The use of block diagrams for the analysis of dynamic problems is well known and widespread. As the systems of interest increase in complexity, new techniques are sought to aid the controls engineer in the analysis and reduction of such systems.

Indeed, at least part of the motivation of modern control theory is to deal with the problem of large system complexities through the organizational features provided by the matrix approach.

The concept of block diagram reduction is to simplify the complex diagram to an equivalent diagram composed of a single block or a few blocks, that is, to topologically

simplify the block diagram.

In reference 1, the predecessor of this effort, a program was evolved which symbolically reduced block diagrams to obtain S form and $i\omega$ form algebraic solutions for desired transfer functions. This approach suffices for intermediate-size diagrams and also gives fully expanded closed-form solutions. However, the method is limited to systems of approximately 30 equations (for transfer functions (G's) which are not too complicated) due to storage limitations (for the IBM 7094 computer). Experience with the predecessor program has shown that while the closed-form solutions are desirable for many scientific endeavors, the bulk of the potential applications were of an engineering nature and could not be performed due to the size limitation. Most of these engineering applications required only numerical results representing the desired transfer function. However, the applications tended to be large and quite complex.

Direct matrix methods (ref. 2), for example, could be applied to these systems. However, these methods would require a complete matrix manipulation for each frequency and would become quite expensive in terms of computer time.

Another possible approach is to use the symbolic techniques as a part of a reduction scheme, together with a decompositional method which would not allow the solution to grow within the computer. The development of this approach is the objective of this study.

The report develops the mathematical concepts and a computer program (REDUCE II) to implement this approach. A numerical evaluation program (EVAL II) to use these symbolic results is also developed. Applications of the programs are presented.

REDUCTION TECHNIQUE

In general, dynamic and control systems are analyzed by considering the characteristics of the elements (individual transfer functions) and combining these elements into a system which is the block diagram. The block diagram elements are (1) summers, which add or subtract two or more signals X's; (2) blocks, which accept one signal and modify it through multiplication by a frequency-sensitive operator G(S); and (3) nodes, which split a signal into two or more parts. These elements are then combined to yield linear block diagrams. For details of block diagrams and their reductions, refer to reference 1. This study addresses itself to the reduction of very large linear block diagrams (of arbitrary topology) to obtain the transfer functions of a desired output to any or all inputs. The symbolic computer language FORMAC (ref. 3) is used to achieve the reduction. The FORMAC language is an experimental extension of FORTRAN IV for the IBM 7094 computer.

Basic Philosophy

The fundamental philosophy of reduction used in the REDUCE II program entails two points which are incorporated to maximize block-diagram-size capability. First, any steps which promote significant expansion of the forms within the computer are avoided. For example, substitution of values of the G's is not done in the reduction program. Second, the reduction of the block diagram must proceed in such a manner as to have the system of equations collapse rather than grow in the computer. That is, as variables are eliminated in the reduction process, the coefficients of the remaining variables become combinations of the original coefficients (G's). As this occurs, these combinations are replaced by single-element coefficients (super G's), and the mathematical equation defined by this substitution is immediately output from the computer.

Technique

The general approach to the reduction is as follows:

(1) The system of linear algebraic equations which represent the block diagram is input to the computer in symbolic form.

(2) Input is an order string, that is, a string of variables which guides the manner in which the reduction is to be performed and indicates when a substitution is to be made. From these data, the program reduces the block diagram by loops and, after each loop is reduced, substitutes a new G (a super G) for combinations of G's formed by the reduction process. By reduction of a loop is meant the elimination of those variables which form a cycle or a circuit in the graph theoretical sense (e.g., see refs. 4 and 5). By virtue of this technique, the system of equations in the computer can only collapse; and hence, the largest system which can be handled is approximately the largest system of equations which can be initially input to the computer, or in practical terms, that set of equations which can be held in core.

From the mathematical point of view then, we are starting with a set of linear equations in the variable X with coefficients (G's) which are considered to be constant. Through the reduction and substitution process, we are transforming the original set of equations into a new set of equations which define the super G's in terms of the original G's and previous super G's and into a single equation involving the X's of the transfer functions.

Even within this framework, a large number of ways exist to proceed, since the form of the equations composing the solution set has not been stipulated. However, the following desirable properties of the solution set can be projected from the intended use:

(1) Since eventually the solution set is to be evaluated numerically for amplitude ratio, phase angle, and so forth, the super G's formed should be nested. That is, a

given super G should depend only on the G's which precede it and on the original G's of the block diagram.

(2) A zero divide occurs when the denominator of a super G is evaluated to be identically zero. The solution set should be of such a form that the likelihood of a zero divide occurring when the solution set is evaluated is minimal.

(3) The solution set should be of such a form that subsequent equation scaling is not necessary. For example, the process could be carried out in such a manner as to allow only super G's of a sum-product form, that is,

$$G_{SUPER} = G_a G_b G_c + G_d G_e G_f + \ldots \text{ etc.}$$

Symbols are defined in appendix A. Now, while this form would be highly desirable in terms of property 2, it leads to a scaling problem. Experience has shown that for certain common block diagram topologies (notably some of the ladder networks) the last super G, even for relatively small diagrams, can be a function of ω to a very large power. This, of course, would only allow evaluation to a very small value of ω without some form of scaling. If scaling is to be done, it should be integral with the formation of the solution set.

(4) Finally, it is desired that the solution set be as compact or concise as possible. This is a purely practical requirement, since for very large block diagrams the volume of punched cards and printout could become unwieldy.

Properties 2 and 3 can be conflicting requirements. Consideration of properties 2 and 3 suggests that a most desirable form for the super G's composing the solution set would be

$$G_{SUPER} = \frac{G_a G_b + G_c G_d \cdot \cdot \cdot}{1 + G_e G_f + G_h G_i G_j \cdot \cdot \cdot} \tag{1}$$

or as close to this form as possible. This form is characterized by the fact that, for G's of the same order, the super G would be of net order 1 and since the denominator has a unity additive term, the possibility of a zero-divide condition (by allowing any G to equal zero) is null. The $G = 0$ special case is considered important since it is the fundamental method of simplifying block diagram topology. It appears that super G's of the form of equation (1) can always be made to occur if the block diagram is reduced by loops.

This is demonstrated by the following simple example. Figure 1 is a block diagram of a loop which is being considered separately from a larger block diagram which contains it. The defining equations for the embedded loop are

$$X_1 - X_2 - X_5 = 0$$

$$X_3 - X_4 - X_6 = 0$$

$$X_3 - G_1X_5 = 0$$

$$X_2 - G_2X_4 = 0$$

Eliminating the loop by eliminating the variables X_4 and X_5 yields

$$X_2 = \frac{G_1G_2}{1 + G_1G_2} X_1 - \frac{G_2}{1 + G_1G_2} X_6$$

and

$$X_3 = \frac{G_1}{1 + G_1G_2} X_1 + \frac{G_1G_2}{1 + G_1G_2} X_6$$

where X_2 and X_3 are the loop outputs defined in terms of X_1 and X_6, the loop inputs. The coefficients of X_1 and X_6 are the super G's associated with the reduction of the loop. These super G's are seen to be of the same form as equation (1).

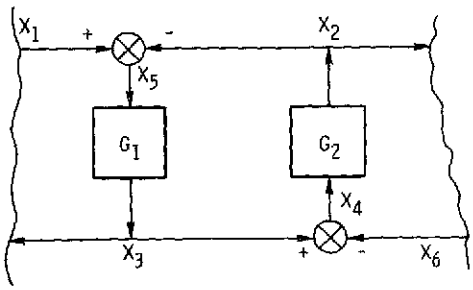


Figure 1. - Block diagram of embedded loop.

The fact that this kind of a solution form is possible, even for a loop embedded in a large block diagram, has been used as a fundamental premise in the structure of the computer program presented in this report. Not every equation affected in the reduction of a loop will contain the 1 + GG + . . . coefficient form. That is, those equations outside the loop, but containing loop variables, will not at this stage of the reduction contain the 1 + GG + . . . form. They might at a later stage. However, postponement of the

super G substitution is not advised, since the coefficients can grow unreasonably large, defeating the program purpose. Contingencies arise when the loops are very large or difficult to identify. These details are considered later.

Program Flow

Before considering the flow of the reduction program it will be helpful to consider the relation between the reduction and evaluation programs. This relation is indicated in figure 2. The flow chart is composed of two basic parts: the algebraic manipulations

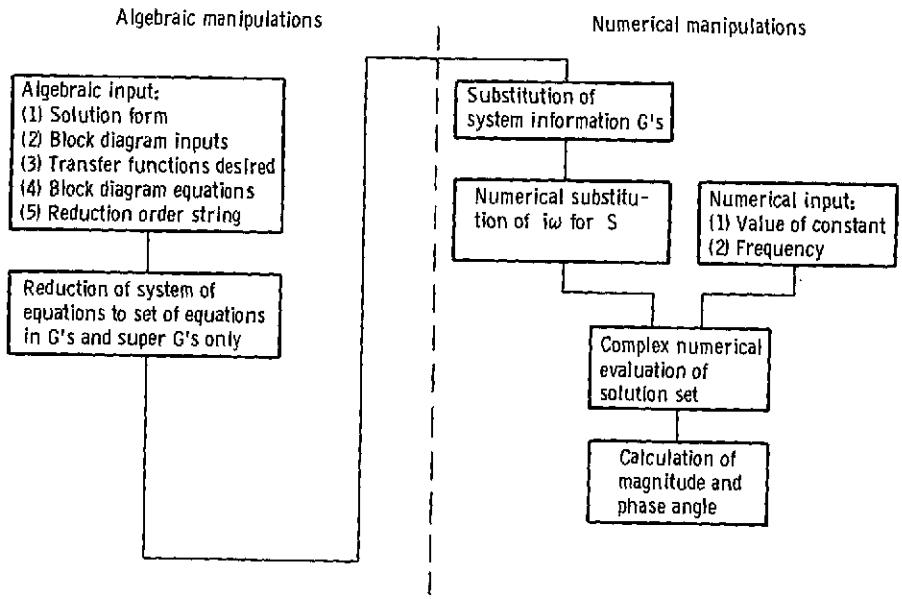


Figure 2. - Relation of algebraic and numerical programs.

are done in FORMAC, and the numerical calculations are done in the companion FORTRAN evaluation program. The expansive step of substituting the values for the G functions into the solution set is postponed until the evaluation program. Thus, larger block diagrams can be reduced. Also flexibility is added since the G's can be changed at will when the numerical evaluations are performed. Hence, it is not necessary to repeat the reduction of a block diagram in order to change the system. The only requirement is that the topology of the block diagram is not changed. Specifically, the topology can be simplified by letting G equal zero, but not made more complex.

The logic and flow of the reduction program are illustrated in the flow chart of figure 3. An understanding of most of the program function is best obtained by referring to a sample reduction as it would be performed by the program. The system to be reduced is shown in block diagram form in figure 4.

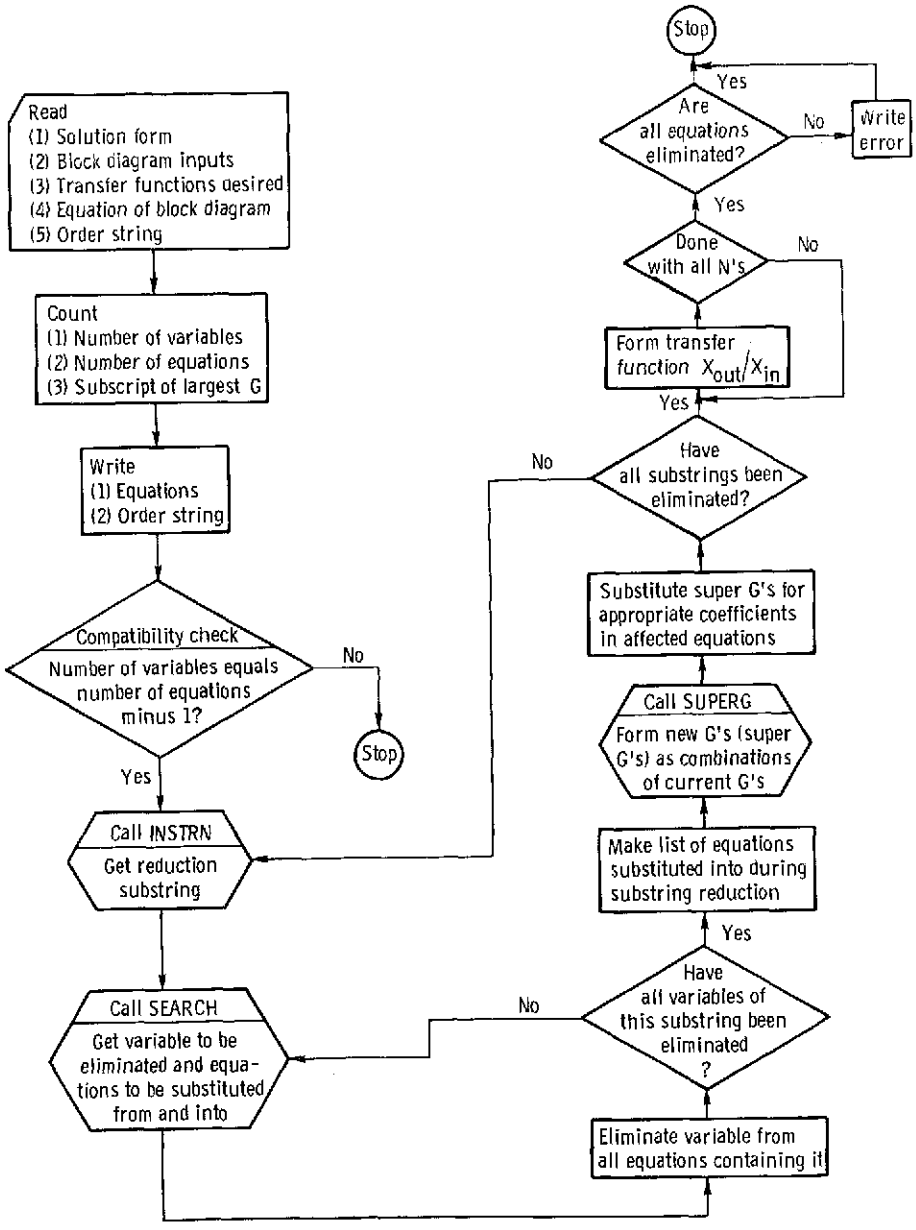


Figure 3. - Reduction program flow chart.

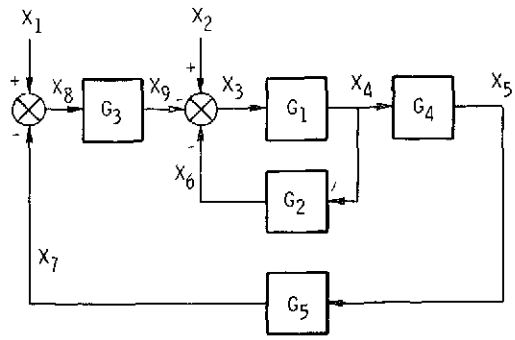


Figure 4. - Block diagram of system to be reduced.

The defining equations for this system are

$$X_1 - X_7 - X_8 = 0 \tag{2}$$

$$X_2 - X_3 - X_6 - X_9 = 0 \tag{3}$$

$$X_4 - G_1X_3 = 0 \tag{4}$$

$$X_6 - G_2X_4 = 0 \tag{5}$$

$$X_9 - G_3X_8 = 0 \tag{6}$$

$$X_5 - G_4X_4 = 0 \tag{7}$$

$$X_7 - G_5X_5 = 0 \tag{8}$$

The inputs are X_1 and X_2, and the desired output is X_5; that is, form the transfer functions X_5/X_1 and X_5/X_2. Since the program reduces by loops, the order string first eliminates the inner loop, that is, the variables X_3 and X_6. After these are eliminated, a super G substitution is made and the next loop (X_4, X_7, X_8, and X_9) is eliminated. The process is continued until all variables but the output and inputs are eliminated. A reasonable order string for this problem then is

$$X_3, X_6, , X_4, X_7, X_8, X_9, ,$$

where the single commas denote a variable elimination and the double commas denote a variable elimination followed by a super G substitution for those coefficients containing more than two G's. A discussion of the order string selection is included in appendix B.

The system of equations is reduced in the following steps in the same manner as

would occur in the program based on the order string given:

STEP 1: Eliminate variable X_3. Since X_3 occurs in equations (3) and (4), the simpler expression (eq. (4)) is substituted into the more complex (eq. (3)). Then, multiplying equation (3) by G_1 and substituting X_4 for G_1X_3 yields

$$G_1X_2 - X_4 - G_1X_6 - G_1X_9 = 0 \tag{3a}$$

$$0 = 0 \tag{4a}$$

STEP 2: Eliminate variable X_6 and make super G substitution. Here X_6 occurs in equations (3a) and (5). Again, results from equation (5) are substituted into equation (3a), giving

$$G_1X_2 - X_4 - G_1G_2X_4 - G_1X_9 = 0 \tag{3b}$$

$$0 = 0 \tag{5b}$$

Combining coefficients yields

$$G_1X_2 - (1 + G_1G_2)X_4 - G_1X_9 = 0 \tag{3c}$$

In making the super G substitution, the coefficients of the X's of all effected equations (here only eq. (3c)) are now searched for a $\pm 1 \pm \left\{ \sum_j \prod_k G \right\}$ form. If the form occurs, the equation is divided through by it and super G's are formed. Hence, equation (3c) becomes

$$G_6X_2 + X_4 - G_6X_9 = 0 \tag{3d}$$

where

$$G_6 = \frac{-G_1}{1 + G_1G_2}$$

is the first super G.

STEP 3: Eliminate variable X_4. Equation (7) is used to eliminate X_4 in equation (3d); then

$$G_4G_6X_2 + X_5 - G_4G_6X_9 = 0 \tag{3e}$$

$$0 = 0 \tag{7a}$$

STEP 4: Eliminate variable X_7. Here X_7 occurs in equations (2) and (8); hence,

$$X_1 - G_5X_5 - X_8 = 0 \tag{2a}$$

$$0 = 0 \tag{8a}$$

STEP 5: Eliminate variable X_8. Using equation (6) in equation (2a) gives

$$G_3X_1 - G_3G_5X_5 - X_9 = 0 \tag{2b}$$

$$0 = 0 \tag{6a}$$

STEP 6: Eliminate variable X_9 and make super G substitution. Substituting from equation (2b) into equation (3e) gives

$$G_4G_6X_2 + X_5 - G_3G_4G_6X_1 + G_3G_4G_5G_6X_5 = 0 \tag{3f}$$

Combining coefficients yields

$$G_3G_4G_6X_1 - G_4G_6X_2 - (1 + G_3G_4G_5G_6)X_5 = 0 \tag{3g}$$

Recognizing the $1\pm$ form and forming the super G's gives

$$G_7X_1 + G_8X_2 + X_5 = 0 \tag{3h}$$

where

$$G_7 = \frac{G_3G_4G_6}{-(1 + G_3G_4G_5G_6)}$$

and

$$G_8 = \frac{-G_4G_6}{-(1 + G_3G_4G_5G_6)}$$

This completes the reduction as indicated by the order string. The only remaining step is to form the desired transfer functions X_5/X_1 and X_5/X_2. To form X_5/X_1, let $X_2 = 0$ in equation (3h); hence,

$$G_7X_1 = -X_5$$

$$\frac{X_5}{X_1} = \frac{G_7}{-1}$$

Likewise for X_5/X_2, let $X_1 = 0$ to give

$$G_8 X_2 = -X_5$$

$$\frac{X_5}{X_2} = \frac{G_8}{-1}$$

While it did not occur in this reduction, if at some step a variable to be eliminated occurs in three or more equations, the simplest equation is used to eliminate it in all equations in which it occurs. Also, if a substitution is called for by the order string and an effected equation does not have a $\left(\pm 1 \pm \sum_j \prod_k G\right)$ form coefficient, a 1.0 can be added to the largest coefficient and this factor used to scale the equation. If this is done, the solution is said to be of the ARTIFICIAL form. If 1.0 is not added but the coefficient itself is used to scale the equation, the solution is called the NATURAL form. The choice of form is made by the user. The details of program implementation are covered in appendix C. The descriptions, listings, and flow charts for the computer programs are detailed in appendix D. A user's manual is provided in appendix E.

Discussion of the companion program EVAL II is also contained in these appendixes.

APPLICATIONS

Applications of the program were chosen to achieve two objectives: first, to study certain operating characteristics of the program, namely, operating time and size capabilities; and second, to demonstrate the program for typical controls block diagrams.

Application 1: Simple Ladder

For this first study, a simple ladder block diagram of the form of figure 5 was reduced. This topology is typical of finite difference approximations to the wave or diffusion equations. For wave equations, this would be a string of coupled oscillators.

The basic equation form for each of the ladder's 600 rungs is

$$X_{n+1} - G_n X_n + G_n X_{n+2} = 0 \qquad \text{for } n = 1, 2, \ldots, 598$$

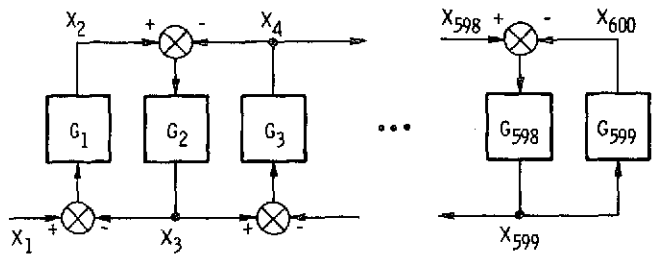


Figure 5. - Block diagram of simple ladder network.

To terminate the ladder, the equation

$$X_{600} - G_{599}X_{599} = 0$$

was used.

The reduction order string for this problem is

$$X_{600},,X_{599},,X_{598},,\ldots X_5,,X_4,,X_3$$

The transfer function desired is that for X_2/X_1. Typical input equations and the first and last parts of the solution set of equations as generated by the program are presented in figure 6. The solution set (fig. 6) is output in the form of FORTRAN expression and not as a mathematical equation. That is, mathematically

$$G(601) = \frac{G_{598}}{DENOM(1)} = \frac{G_{598}}{-G_{598}G_{599} - 1.0}$$

The time history of reduction of the 600-equation set is shown in figure 7. As might be expected, the slope of the curve indicates that the reduction proceeds more quickly as variables are eliminated. The total reduction time for the 600-rung ladder is about 3500 seconds.

The repetitive nature of block diagram topology allows the solution to be readily checked. Also, the repetition, together with the order of solution chosen, allows one to consider the problem as a new ladder one rung shorter after each variable is eliminated Hence, it could appear as though 600 ladder networks of lengths 1 to 600 were reduced. The initial rate of reduction of each of these ladders against the ladder size is shown in figure 8. Here the rates vary from 0.1 variable per second for the 600-equation ladder to 0.5 variable per second for a 25-equation ladder.

The 600-equation ladder is the practical upper limit for the computing system used for this study. Larger problems could be input; however, peripheral storage (noncore) would be required, thus heavily penalizing computing time.

```
THE BLOCK DIAGRAM INPUTS

     X(1)$

THE SOLUTION IS FOR X(2)$  TO  X(1)$

THE BLOCK DIAGRAM EXPRESSIONS

      1  X(   2)-G(   1)*X(   1)+G(   1)*X(   3)$
      2  X(   3)-G(   2(*X(   2)+G(   2)*X(   4)$
      3  X(   4)-G(   3)*X(   3)+G(   3)*X(   5)$
      4  X(   5)-G(   4)*X(   4)+G(   4)*X(   6)$
      5  X(   6)-G(   5)*X(   5)+G(   5)*X(   7)$
      6  X(   7)-G(   6)*X(   6)+G(   6)*X(   8)$
      7  X(   8)-G(   7)*X(   7)+G(   7)*X(   9)$
      8  X(   9)-G(   8)*X(   8)+G(   8)*X(  10)$
      9  X(  10)-G(   9)*X(   9)+G(   9)*X(  11)$
     10  X(  11)-G(  10)*X(  10)+G(  10)*X(  12)$
     11  X(  12)-G(  11)*X(  11)+G(  11)*X(  13)$
     12  X(  13)-G(  12)*X(  12)+G(  12)*X(  14)$
     13  X(  14)-G(  13)*X(  13)+G(  13)*X(  15)$
     14  X(  15)-G(  14)*X(  14)+G(  14)*X(  16)$
     15  X(  16)-G(  15)*X(  15)+G(  15)*X(  17)$
     16  X(  17)-G(  16)*X(  16)+G(  16)*X(  18)$
     17  X(  18)-G(  17)*X(  17)+G(  17)*X(  19)$
     18  X(  19)-G(  18)*X(  18)+G(  18)*X(  20)$

                .     .     .     .     .

                .     .     .     .     .

    593  X(  594)-G(  593)*X(  593)+G(  593)*X(  595)$
    594  X(  595)-G(  594(*X(  594)+G(  594)*X(  596)$
    595  X(  596)-G(  595)*X(  595)+G(  595)*X(  597)$
    596  X(  597)-G(  596)*X(  596)+G(  596)*X(  598)$
    597  X(  598)-G(  597)*X(  597)+G(  597)*X(  599)$
    598  X(  599)-G(  598)*X(  598)+G(  598)*X(  600)$
    599  X(  600)-G(  599)*X(  599)$$
```

(a) Input expressions.

```
* * * OUTPUT FROM REDUCE II * * *

DENOM(   1)=
              -G(598)*G(599)-1.0

G(   600)=G(598)

G(   600)=G( 600)/DENOM(   1)

DENOM(   2)=
              G(597)*G(600)-1.0

G(   601)=G(597)

G(   601)=G( 601)/DENOM(   2)

DENOM(   3)=
              G(596)*G(601)-1.0

                .   .   .

                .   .   .

                .   .   .
DENOM(   597)=
              G(2)*G(1195)-1.0
G(   1196)=G(2)

G(1196)=G(   1196)/DENOM( 597)

DENOM(   598)=
              G(1)*G(1196)-1.0

G(   1197)=G(1)

G(   1197)=G(1197)/DENOM( 598)

     TRANSFER FUNCTION X(   2)/X(   1)

          THE NUMERATOR
              G(1197)

          THE DENOMINATOR
              (-1.0)
```

(b) Solution-set equations.

Figure 6. - Simple-ladder input and solution set.

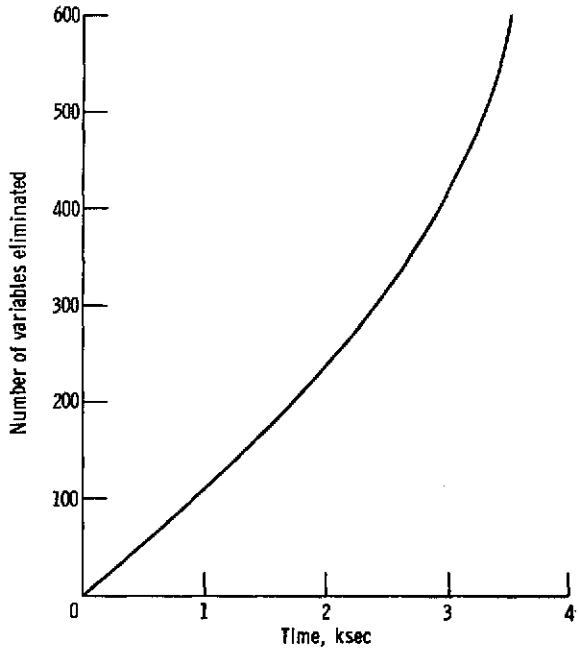


Figure 7. - Reduction history of 600-equation simple ladder.

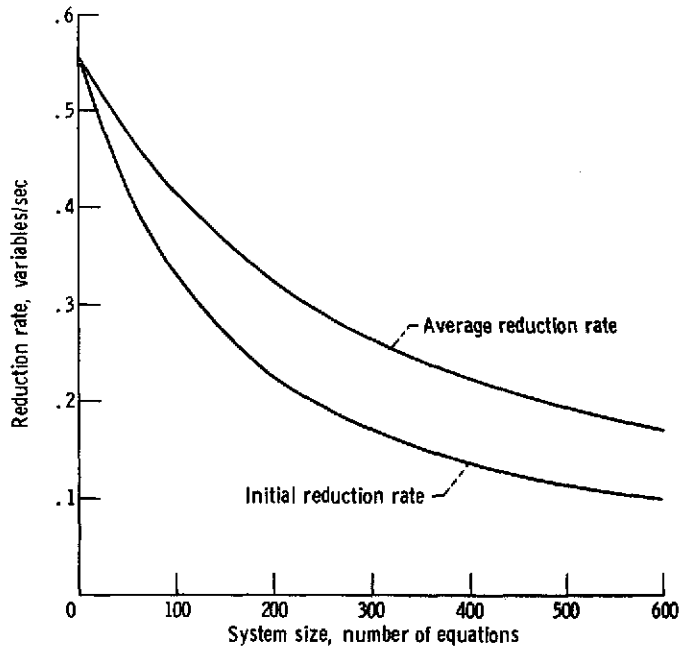


Figure 8. - Reduction rate as function of problem size for 600-equation simple ladder.

Application 2: Complex Ladder

For this application, a ladder of coupled four-terminal networks was reduced. The ladder is composed of 150 rungs, each having two equations; the block diagram is shown in figure 9. For a typical application, each rung of such a ladder would represent a distributed parameter representation of the wave equation. That is, the transfer functions of the blocks would be complex hyperbolic functions or delay times. The topology has wide application.

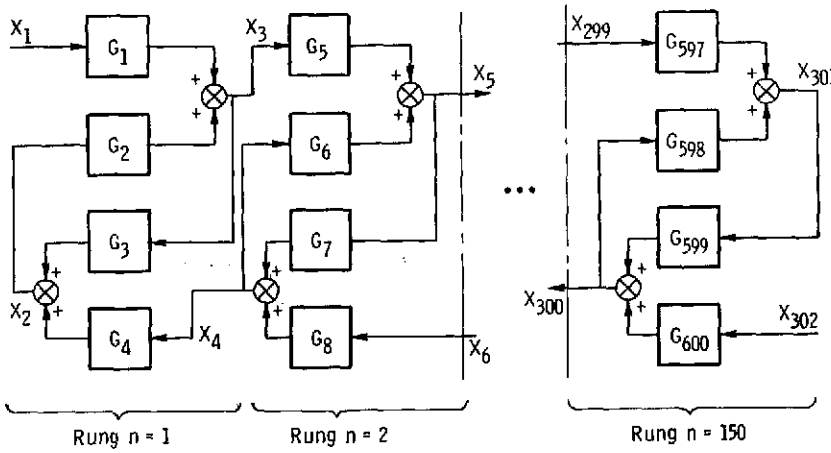


Figure 9. - Block diagram of complex ladder network (four-terminal elements).

The basic equations for each rung of the ladder, for $n = 1, 2, \ldots, 150$, are

$$X_{2n+1} - G_{4n-3}X_{2n-1} - G_{4n-2}X_{2n} = 0$$

and

$$X_{2n} - G_{4n-1}X_{2n+1} - G_{4n}X_{2n+2} = 0$$

The desired transfer functions are X_2/X_1 and X_2/X_{302}. The reduction order is $X_{301},, X_{300}, X_{299},, X_{298}, X_{297},, \ldots,, X_6, X_5,, X_4, X_3$.

The first and last parts of the solution-set input equations are presented in figure 10 The history of the reduction is shown in figure 11. The total time for the reduction is 1010 seconds.

Somewhat larger ladders of this type could be reduced with the program if the dimension of G is increased. This was not attempted in this study.

```
THE BLOCK DIAGRAM INPUTS

     X(1)$
     X(302)

THE SOLUTION IS FOR X(2)$  TO THE INPUTS

THE BLOCK DIAGRAM EXPRESSIONS

      1  X(   3)-G(   1)*X(   1)-G(   2)*X(   2 $
      2  X(   2)-G(   3)*X(   3)-G(   4)*X(   4)$
      3  X(   5)-G(   5)*X(   3)-G(   6)*X(   4)$
      4  X(   4)-G(   7)*X(   5)-G(   8)*X(   6)$
      5  X(   7)-G(   9)*X(   5)-G(  10)*X(   6)$
      6  X(   6)-G(  11)*X(   7)-G(  12)*X(   8)$
      7  X(   9)-G(  13)*X(   7)-G(  14)*X(   8)$
      8  X(   8)-G(  15)*X(   9)-G(  16)*X(  10)$
      9  X(  11)-G(  17)*X(   9)-G(  18)*X(  10)$
     10  X(  10)-G(  19)*X(  11)-G(  20)*X(  12)$
     11  X(  13)-G(  21)*X(  11)-G(  22)*X(  12)$
     12  X(  12)-G(  23)*X(  13)-G(  24)*X(  14)$
     13  X(  15)-G(  25)*X(  13)-G(  26)*X(  14)$
     14  X(  14)-G(  27)*X(  15)-G(  28)*X(  16)$
     15  X(  17)-G(  29)*X(  15)-G(  30)*X(  16)$
     16  X(  16)-G(  31)*X(  17)-G(  32)*X(  18)$
     17  X(  19)-G(  33)*X(  17)-G(  34)*X(  18)$
     18  X(  18)-G(  35)*X(  19)-G(  36)*X(  20)$

    292  X(292)-G(583)*X(293)-G(584)*X(294)$
    293  X(295)-G(585)*X(293)-G(586)*X(294)$
    294  X(294)-G(587)*X(295)-G(588)*X(296)$
    295  X(297)-G(589)*X(395)-G(590)*X(296)$
    296  X(296)-G(591)*X(297)-G(592)*X(298)$
    297  X(299)-G(593)*X(297)-G(594)*X(298)$
    298  X(298)-G(595)*X(299)-G(596)*X(300)$
    299  X(301)-G(597)*X(299)-G(598)*X(300)$
    300  X(300)-G(599)*X(301)-G(600)*X(302)$
```

(a) Input expressions.

```
* * * OUTPUT FROM REDUCE 11 * * *

DENOM(   1)=
              G(598)*G(599)-1.0

G(   601)=G(597)*G(599)

G(   601)=G(  601)/DENOM(   1)

G(   602)=G(600)

G(   602)=G(  602)/DENOM(   1)

DENOM(   2)=
              G(594)*G(595)-G(594)*G(596)*G(601)-1.0

G(   603)=G(593)*G(595)-G(593)*G(596)*G(601)

G(   603)=G(  603)/DENOM(   2)

                .   .   .   .

     DENOM(   150)=
                   G(2)*G(3)-G(2)*G(4)*G(897)-1.0

     G(   899)=G(1)*G(3)-G(1)*G(4)*G(897)
                )
     G(   899)=G(   899)/DENOM(   150)

     G(   900)=-G(4)*G(898)

     G(   900)=G(   900)/DENOM(   150)

          TRANSFER FUNCTION X(   2)/X(   1)

               THE NUMERATOR
                   G(899)

               THE DENOMINATOR
                   (-1.0)

          TRANSFER FUNCTION X(   2)/X(  302)

               THE NUMERATOR
                   G( 900)

               THE DENOMINATOR
                   (-1.0)
```

(b) Solution-set equations.

Figure 10. - Complex-ladder input and solution set.

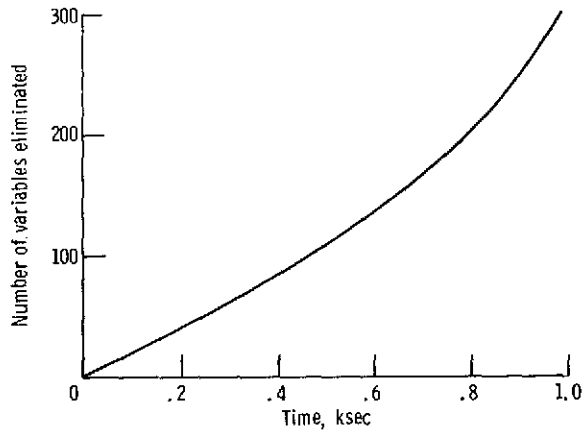


Figure 11. - Reduction history of 300-equation complex ladder.

Application 3: Jet-Engine-Inlet Shock-Position Control

The block diagram of figure 12 represents a system for control of the position of a shock wave in a supersonic inlet for a jet engine using engine fuel flow. "Modern" control techniques were used to design a controller to minimize the occurrence of inlet unstarts in the presence of a stochastic disturbance. The control designer wished to obtain specific transfer functions from which system frequency domain behavior could be studied. Since a PADE' approximation to the inlet plant dead time had been made for control design purposes, it was important to determine the effect of this approximation on dynamic performance. Transfer functions X_7/X_1, X_7/X_2, and X_7/X_3 were obtained for this reason. The equations representing the plant and controller (the block diagram) as input to the program are shown in figure 13(a). Also shown is the order of reduction.

The block diagram inputs are X_1, X_2, and X_3. The results of this reduction (using the ARTIFICIAL form of solution) are given in figure 13(b). From these results, evaluations with different plants and controllers can be made as long as the topology of the block diagram does not become more complex. Essentially, any simplification of the topology is possible.

For example, we can check the plant transfer function by simply evaluating X_7/X_2 with G_{16} to G_{21} equal to zero. This check capability, by specifying certain G's to be zero, is useful when working with very large systems. Also, it is nearly always possible when the solution is of the ARTIFICIAL form.

The time required to execute this reduction is 25 seconds. The system is approximately equivalent to 50 elementary equations as determined by consideration of the number of simple summers and blocks of the block diagram.

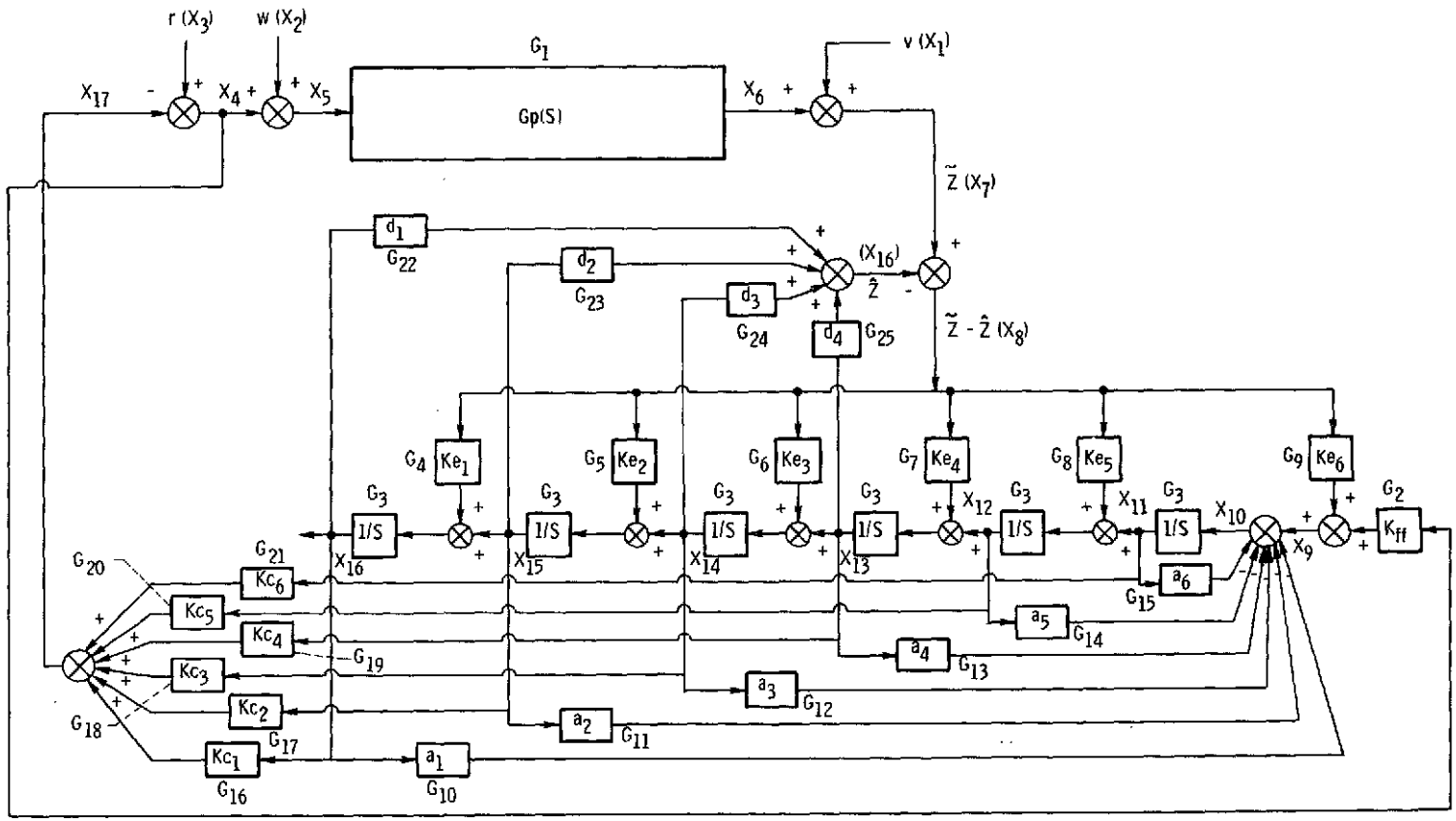


Figure 12. - Block diagram of jet engine inlet control.

```
THE BLOCK DIAGRAM INPUTS

    X(1)$
    X(2)$
    X(3)$

THE SOLUTION IS FOR X(7)$  TO THE INPUTS

THE BLOCK DIAGRAM EXPRESSIONS

     1  -X(10)+X(9)-G(10)*X(16)-G(11)*X(15)-G(12)*X(14)-G(13)*X(13)-G(14)*X(12)-G(15)*X(
     1  11)$
     2  X(1)+X(6)-X(7)$
     3  X(3)-X(17)-X(4)$
     4  X(2)+X(4)-X(5)$
     5  X(6)-G(1)*X(5)$
     6  X(7)-X(18)-X(8)$
     7  -X(9)+G(9)*X(8)+G(2)*X(4)$
     8  X(11)-G(3)*X(10)$
     9  -X(12)+G(3)*G(8)*X(8)+G(3)*X(11)$
    10  -X(13)+G(3)*G(7)*X(8)+G(3)*X(12)$
    11  -X(14)+G(3)*G(6)*X(8)+G(3)*X(13)$
    12  -X(15)+G(3)*G(5)*X(8)+G(3)*X(14)$
    13  -X(16)+G(3)*G(4)*X(8)+G(3)*X(15)$
    14  -X(17)+G(16)*X(16)+G(17)*X(15)+G(18)*X(14)+G(19)*X(13)+G(20)*X(12)+G(21)*X(11)$
    15  -X(18)+G(22)*X(16)+G(23)* X(15)+G(24)*X(14)+G(25)*X(13)$$

THE REDUCTION ORDER IS

     1  X(10),X(11),,X(12),,X(13),,X(14),,X(15),,X(16),,X(9),X(18),X(8),,X(6),X(17),X(4)
     2  ,,X(5)$

THE LARGEST VARIABLE SUBSCRIPT IS   18
THE NUMBER OF EXPRESSIONS IS   15
THE LARGEST FUNCTION (G) SUBSCRIPT IS  25
```

(a) Input equations.

```
          * * * OUTPUT FROM REDUCE II * * *


DENOM(    1)=
                   -G(3)*G(20)-G(21)+1.0


G(   26)=
                   -G(3)*G(20)-G(21)
G(   26)=G(   26)/DENOM(    1)


G(   27)=
                    G(3)*G(8)*G(21)
G(   27)=G(   27)/DENOM(    1)


G(   28)=
                   -G(3)*G(16)
G(   28)=G(   28)/DENOM(    1)


G(   29)=
```

(b) Solution-set equations.

Figure 13. - Input and solution set for jet-engine-control problem.

```
                -G(3)*G(17)
G(  29)=G(  29)/DENOM(   1)

G(  30)=
                -G(3)*G(18)
G(  30)=G(  30)/DENOM(   1)

G(  31)=
                -G(3)*G(19)
G(  31)=G(  31)/DENOM(   1)

G(  32)=
                 G(3)
G(  32)=G(  32)/DENOM(   1)

DENOM(   2)=
                 G(3)*G(15)+G(3)**2  *G(14)+1.0

G(  33)=
                -G(3)*G(8)-G(3)**2  *G(8)*G(15)
G(  33)=G(  33)/DENOM(   2)

G(  34)=
                 G(3)**2  *G(10)
G(  34)=G(  34)/DENOM(   2)

G(  35)=
                 G(3)**2  *G(11)
G(  35)=G(  35)/DENOM(   2)

G(  36)=
                 G(3)**2  *G(12)
G(  36)=G(  36)/DENOM(   2)

G(  37)=
                 G(3)**2  *G(13)
G(  37)=G(  37)/DENOM(   2)

G(  38)=
                -G(3)**2
G(  38)=G(  38)/DENOM(   2)

DENOM(   3)=
                 G(3)*G(7)*G(26)-G(3)*G(27)+1.0

G(  39)=
                 G(3)*G(7)*G(26)-G(3)*G(27)
G(  39)=G(  39)/DENOM(   3)
```

(b) Continued.

Figure 13. - Continued.

```
G(  40)=
                -G(3)*G(28)
G(  40)=G(  40)/DENOM(   3)

G(  41)=
                -G(3)*G(29)
G(  41)=G(  41)/DENOM(   3)

G(  42)=
                -G(3)*G(30)
G(  42)=G(  42)/DENOM(   3)

G(  43)=
                -G(3)*G(31)-G(26)
G(  43)=G(  43)/DENOM(   3)

G(  44)=
                -G(3)*G(32)
G(  44)=G(  44)/DENOM(   3)

DENOM(   4)=
                -G(3)*G(37)-1.0

G(  45)=
                G(3)*G(7)-G(3)*G(33)
G(  45)=G(  45)/DENOM(   4)

G(  46)=
                -G(3)*G(34)
G(  46)=G(  46)/DENOM(   4)

G(  47)=
                -G(3)*G(35)
G(  47)=G(  47)/DENOM(   4)

G(  48)=
                -G(3)*G(36)
G(  48)=G(  48)/DENOM(   4)

G(  49)=
                -G(3)*G(38)
G(  49)=G(  49)/DENOM(   4)

DENOM(   5)=
                -G(3)*G(24)-G(25)+1.0

G(  50)=
```

(b) Continued.

Figure 13. - Continued.

```
                -G(3)*G(24)-G(25)
G(   50)=G(   50)/DENOM(    5)


G(   51)=
                G(3)*G(6)*G(25)
G(   51)=G(   51)/DENOM(    5)


G(   52)=
                -G(3)*G(22)
G(   52)=G(   52)/DENOM(    5)


G(   53)=
                -G(3)*G(23)
G(   53)=G(   53)/DENOM(    5)


G(   54)=
                G(3)
G(   54)=G(   54)/DENOM(    5)


DENOM(    6)=          1
              G(3)*G(6)*G(43)-G(3)*G(39)+1.0


G(   55)=
                G(3)*G(6)*G(43)-G(3)*G(39)
G(   55)=G(   55)/DENOM(    6)


G(   56)=
                -G(3)*G(40)
G(   56)=G(   56)/DENOM(    6)


G(   57)=
                -G(3)*G(41)
G(   57)=G(   57)/DENOM(    6)


G(   58)=
                -G(3)*G(42)-G(43)
G(   58)=G(   58)/DENOM(    6)


G(   59)=
                -G(3)*G(44)
G(   59)=G(   59)/DENOM(    6)


DENOM(    7)=
                -G(3)*G(48)-1.0


G(   60)=
                G(3)*G(6)-G(3)*G(45)
G(   60)=G(   60)/DENOM(    7)
```

(b) Continued.

Figure 13. - Continued.

```
G(  61)=
                -G(3)*G(46)
G(  61)=G(  61)/DENOM(   7)


G(  62)=
                -G(3)*G(47)
G(  62)=G(  62)/DENOM(   7)


G(  63)=
                -G(3)*G(49)
G(  63)=G(  63)/DENOM(   7)


DENOM(   8)=
                G(3)*G(5)*G(50)-G(3)*G(51)+1.0


G(  64)=
                G(3)*G(5)*G(50)-G(3)*G(51)
G(  64)=G(  64)/DENOM(   8)


G(  65)=
                -G(3)*G(52)
G(  65)=G(  65)/DENOM(   8)


G(  66)=
                -G(3)*G(53)-G(50)
G(  66)=G(  66)/DENOM(   8)


G(  67)=
                -G(3)*G(54)
G(  67)=G(  67)/DENOM(   8)


DENOM(   9)=
                G(3)*G(5)*G(58)-G(3)*G(55)+1.0


G(  68)=
                G(3)*G(5)*G(58)-G(3)*G(55)
G(  68)=G(  68)/DENOM(   9)


G(  69)=
                -G(3)*G(56)
G(  69)=G(  69)/DENOM(   9)


G(  70)=
                -G(3)*G(57)-G(58)
G(  70)=G(  70)/DENOM(   9)


G(  71)=
```

(b) Continued.

Figure 13. - Continued.

```
                -G(3)*G(59)
G(  71)=G(  71)/DENOM(   9)

DENOM(  10)=
                -G(3)*G(62)-1.0

G(  72)=
                 G(3)*G(5)-G(3)*G(60)
G(  72)=G(  72)/DENOM(  10)

G(  73)=
                -G(3)*G(61)
G(  73)=G(  73)/DENOM(  10)

G(  74)=
                -G(3)*G(63)
G(  74)=G(  74)/DENOM(  10)

DENOM(  11)=
                 G(3)*G(4)*G(66)-G(3)*G(64)+1.0

G(  75)=
                 G(3)*G(4)*G(66)-G(3)*G(64)
G(  75)=G(  75)/DENOM(  11)

G(  76)=
                -G(3)*G(65)-G(66)
G(  76)=G(  76)/DENOM(  11)

G(  77)=
                -G(3)*G(67)
G(  77)=G(  77)/DENOM(  11)

DENOM(  12)=
                 G(3)*G(4)*G(70)-G(3)*G(68)+1.0

G(  78)=
                 G(3)*G(4)*G(70)-G(3)*G(68)
G(  78)=G(  78)/DENOM(  12)

G(  79)=
                -G(3)*G(69)-G(70)
G(  79)=G(  79)/DENOM(  12)

G(  80)=
                -G(3)*G(71)
G(  80)=G(  80)/DENOM(  12)
```

(b) Continued.

Figure 13. - Continued.

```
DENOM(  13)=
                -G(3)*G(73)-1.0

G(  81)=
                G(3)*G(4)-G(3)*G(72)
G(  81)=G(  81)/DENOM(  13)

G(  82)=
                -G(3)*G(74)
G(  82)=G(  82)/DENOM(  13)

DENOM(  14)=
                -G(75)+G(76)*G(81)+1.0

G(  83)=
                -G(75)+G(76)*G(81)
G(  83)=G(  83)/DENOM(  14)

G(  84)=
                G(76)*G(82)
G(  84)=G(  84)/DENOM(  14)

G(  85)=
                -G(77)
G(  85)=G(  85)/DENOM(  14)

DENOM(  15)=
                -G(78)+G(79)*G(81)+1.0

G(  86)=
                -G(78)+G(79)*G(81)
G(  86)=G(  86)/DENOM(  15)

G(  87)=
                G(79)*G(82)
G(  87)=G(  87)/DENOM(  15)

G(  88)=
                -G(80)
G(  88)=G(  88)/DENOM(  15)

DENOM(  16)=
                G(2)*G(83)*G(87)-G(2)*G(84)*G(86)-G(2)*G(85)*G(87)+1.0

G(  89)=
                G(2)*G(83)*G(87)-G(2)*G(84)*G(86)-G(2)*G(85)*G(87)
G(  89)=G(  89)/DENOM(  16)
```

(b) Continued.

Figure 13. - Continued.

```
G(  90)=
                G(9)*G(84)*G(88)+G(83)*G(88)-G(85)*G(88)
G(  90)=G(  90)/DENOM(  16)

G(  91)=
                -G(9)*G(85)*G(87)-G(85)*G(86)
G(  91)=G(  91)/DENOM(  16)

DENOM(  17)=
                G(89)-G(90)+1.0

G(  92)=
                G(89)-G(90)
G(  92)=G(  92)/DENOM(  17)

G(  93)=
                -G(89)+G(90)
G(  93)=G(  93)/DENOM(  17)

G(  94)=
                -G(90)
G(  94)=G(  94)/DENOM(  17)

G(  95)=
                -G(91)
G(  95)=G(  95)/DENOM(  17)

DENOM(  18)=
                G(1)*G(95)+G(93)+1.0

G(  96)=
                G(1)*G(95)+G(93)
G(  96)=G(  96)/DENOM(  18)

G(  97)=
                G(1)*G(92)
G(  97)=G(  97)/DENOM(  18)

G(  98)=
                G(1)*G(94)
G(  98)=G(  98)/DENOM(  18)

G(  99)=
                -G(93)
G(  99)=G(  99)/DENOM(  18)

    TRANSFER FUNCTION X(  7)/X(  1)
```

(b) Continued.

Figure 13. - Continued.

```
THE NUMERATOR
    G(99)

THE DENOMINATOR
    -G(96)

TRANSFER FUNCTION X(  7)/X(  2)

THE NUMERATOR
    G(97)

THE DENOMINATOR
    -G(96)

TRANSFER FUNCTION X(  7)/X(  3)

THE NUMERATOR
    G(98)

THE DENOMINATOR
    -G(96)
```

(b) Concluded.

Figure 13. - Concluded.

A numerical evaluation for the following set of G's (system information) was made by using the EVAL program:

$$G_1 = \frac{846.7(-0.833\times10^{-2}\,S^3 + 0.1\,S^2 - 0.5\,S + 1.0)}{S^6 + 18.28\,S^5 + 155.4\,S^4 + 743.9\,S^3 + 2038S^2 + 2823S + 846.7}$$

$$G_2 = 846.7$$

$$G_3 = \frac{1}{S}$$

$$G_4, G_5, \ldots, G_9 = 2.041662,\ -0.3595,\ -2.54,\ 8.723,\ -1.5025,\ -146.932$$

$$G_{10}, G_{11}, \ldots, G_{15} = 846.7,\ 2823,\ 2038,\ 743.9,\ 155.4,\ 18.28$$

$$G_{16}, G_{17}, \ldots, G_{21} = 998.9,\ 600.97,\ 155.18,\ 20.745,\ 1.300,\ 0.038$$

For the nonapproximate case, the plant transfer function becomes

$$G_1 = \frac{7.06\,e^{-1.0S}}{S^3 + 6.33\,S^2 + 20.01\,S + 7.06}$$

The frequency range of interest is 0.01 to 4.0 hertz, in steps of 0.01 hertz. (Because of time and frequency scaling requirements of the user, frequency was multiplied

by 100.) The execution time required to evaluate these 400 points plus an additional 600 points for the automated plot routine for all three transfer functions is 32 seconds. Plots of typical results for the X_7/X_1 transfer function are presented in figure 14. The difference in the plots is of course the effect of the PADE' approximation at the higher frequencies.

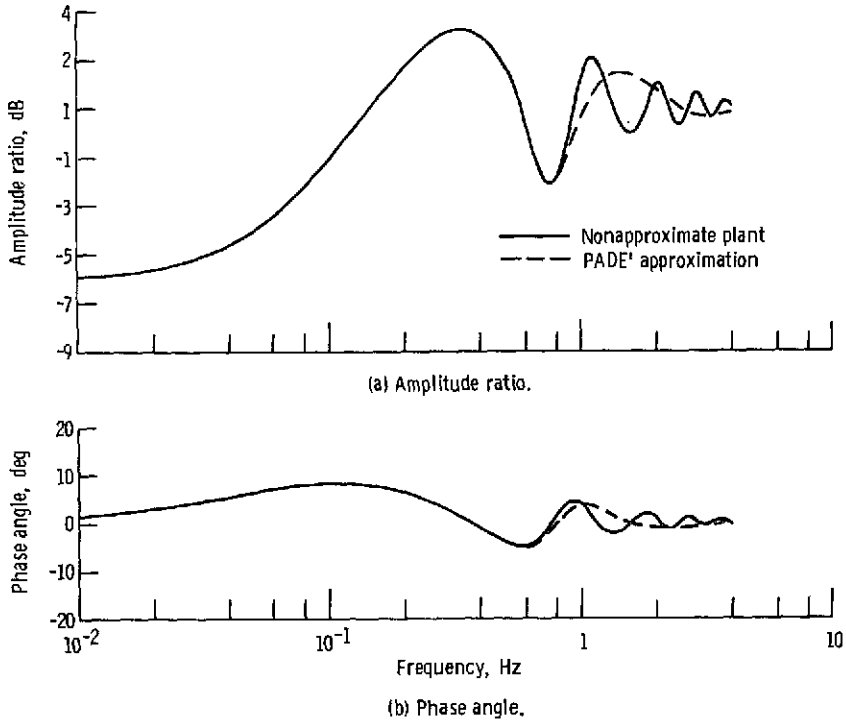


Figure 14. - Typical transfer function results, for jet-engine-control problem.

Applications: Some Comments

Execution time. - A summary plot of reduction rate as a function of problem size for the applications studied is presented in figure 15. Generally, the program reduces more quickly for smaller problem size, as would be expected.

Storage. - Four terms are involved in considerations of storage for this program; they are related by the equation

PROGRAM STORAGE + INITIAL EQUATION STORAGE

$$+ X_{STORAGE} + G_{STORAGE} = CONSTANT$$

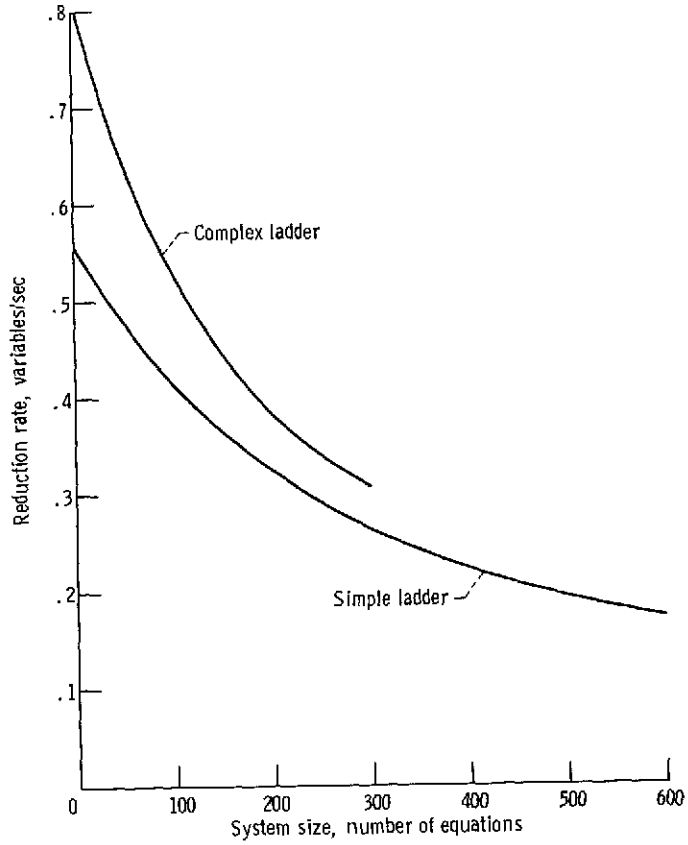


Figure 15. - Comparison of reduction rate as function of problem size for complex and simple ladders.

Since program storage is fixed when considering problem capability for a given computer, we have

$$\text{INITIAL EQUATION STORAGE} + X_{\text{STORAGE}} + G_{\text{STORAGE}} = \text{CONSTANT}$$

Now for large problems with relatively few inputs

$$N_{EQ} = \text{NUMBER OF EQUATIONS} \approx \text{NUMBER OF X's} = N_X$$

and for a given class of problem we can write

$$X_{\text{STORAGE}} = K' N_{EQ}$$

Also, it is clear that

$$\text{INITIAL EQUATION STORAGE} = K'' N_{EQ}$$

$$G_{STORAGE} = K'''N_G$$

Combining these results yields

$$K'N_{EQ} + K''N_{EQ} + K'''N_G = CONSTANT$$

Hence,

$$N_{EQ} + KN_G = CONSTANT$$

Here N_{EQ} is the original number of equations and N_G is the total number of G's (original plus super G's). This relation then should be useful in setting or changing array sizes for either (1) different computers or (2) problems where the relation of N_G to N_{EQ} varies substantially.

CONCLUSIONS

Techniques have been established and a computer program written to allow the computer reduction of large arbitrary block diagrams. The techniques used involve symbolic manipulation of the defining equations by means of the FORMAC symbolic computer language. With the program, transfer functions can be determined for any output variable to any or all input variables. Since the primary objective of the work was to handle quite large and complex block diagrams, the solution is not formed as a single relationship for a desired transfer function. The solution, instead, is determined as a set of algebraic equations. Each of these equations, called super G's, is defined in terms of the original system G's and the super G's which precede it.

The transfer functions are represented as a set of nested equations. This form has the advantage of being considerably more compact than the single relationship. Also, since the solution-set relations can be output from the computer as they are formed, the critical problem of computer storage is alleviated. Further, the nested form of the solution set allows straightforward numerical evaluation.

The program is premised on the reduction of the block diagram by loops. When this is achieved, the solution is automatically scaled so that effective large powers of frequency do not occur. Also, the denominators are such that the possibility of zero-divides is small.

Two forms of the solution set are possible at the option of the user.

A program has been written to evaluate the solution set. This evaluation yields the real and imaginary parts and the phase angle and magnitude and log magnitude ratios of the transfer functions.

The advantages of the technique developed in the present study over previous techniques are as follows:

1. As in the predecessor program, the arrangement of the input data in the form of arrays and the matrix manipulation for each frequency point are not required. (The symbolic solution set need be generated only once and then is reevaluated for each frequency.)

2. Unlike the predecessor program, the generation of a solution set instead of a single function allows the solution of considerably more complex problems, that is, systems involving as many as 600 equations and a total of 1200 G's.

The user input to the reduction program consists of two basic pieces of information: (1) the algebraic equations representing the block diagram; and (2) the order string, which controls the manner in which the reduction is performed. The first part represents a minimal effort on the part of the user. The determination of the order string does pose a small burden for the user. However, usually the consequence of a poor choice for the order string is a relatively small increase in the size of the solution set.

The computing time required for the program varies with the size of the problem being solved. Typical rates are 0.6 variable per second for small problems to an initial rate of 0.1 variable per second for the larger problems.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, October 5, 1973,
501-24.

APPENDIX A

SYMBOLS

DENOM	denominator of super G
G	transfer function for block in original block diagram
G_{SUPER}	function defining part of solution set
i	$\sqrt{-1}$
K	constant
$N_{( )}$	number of ( )
S	Laplace variable
X	signal or variable in a block diagram
ω	circular frequency, rad/sec
Subscripts:	
a, b, c, . . . , j, k, n	index numbers 1, 2, . . .
EQ	equations
i	inputs
os	order string
v	variables

APPENDIX B

ORDER STRING

The order string is a list of the block diagram variables. The order of the variables occurring in the list is the order in which they are to be reduced in the program. The variables are separated by commas. For example, X_a, X_b, X_c indicates eliminate variable X_a, then X_b, then X_c. Separation of a pair of variables by double commas indicates that a super G substitution is to be made at that point.

The program logic is premised on reduction of the block diagram by loops. The loops of the block diagram are analogous to the cycles or circuits of graph theory (refs. 4 and 5). That is, a loop is a progression of variables in a block diagram that closes on itself. Variables which split off (nodes) and leave the loop or those which enter the loop (through a summer) will be called communicating spurs, or simply spurs.

The suggested best approach then will be to identify from the block diagram those loops having the least number of communicating spurs. The variables composing the loop, less the communicating spur variables, if possible, then form an order substring. The order of reduction of the variables forming the loop does not appear to be important. In the example of figure 16, the loop comprised of X_3, X_4, X_{10} has only two spurs,

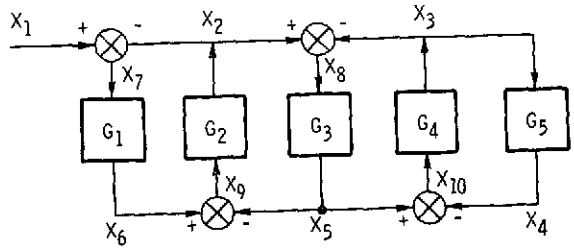


Figure 16. - Block diagram used to show reduction of order string.

namely X_5 and X_3. Hence, an appropriate substring would be X_4, X_{10} or X_{10}, X_4. Elimination of this loop would leave a new loop with variables X_5, X_3, and X_8 with spurs X_5 and X_2 giving a substring X_8, X_3. After the reduction of each loop, a super G substitution should be made, that is, separation by double commas. Hence, the order string for this reduction might be

$$X_{10}, X_4, , X_3, X_8, , X_5, X_9, , X_2, X_7$$

where we have considered X_6 to be the output variable.

While it is not always possible to get an end loop, that is, one with only two spurs,

the program appears to be quite forgiving; and even taking internal loops as shown in the text, gives rise to the same general forms. This will generate a larger number of super G's, however.

Several checks can be made after an order string has been selected:

(1) All variables in the problem must be represented in the order string except the output and the inputs (which, of course, are not eliminated in the reduction).

(2) No variable can be represented more than once.

(3) From algebraic theory,

$$N_v - N_{EQ} = N_i$$

for the original system of equations. And, for the order string,

$$(N_v)_{os} = N_v - N_i - 1$$

Hence

$$(N_v)_{os} = N_{EQ} - 1$$

This check is also made by the program.

The choice of the order string for very large or very complex (interactive) diagrams may be difficult. Occasionally, for such cases the solution generated will be valid only over some limited frequency band. This is usually recognized by an instability occurring in the numerical solution above some frequency. In such cases, it may be necessary to change the order string to achieve a better solution form.

APPENDIX C

PROGRAM IMPLEMENTATION

The complete reduction and numerical evaluation of a block diagram is done in two separate programs. The reduction program is called REDUCE II and is written in FORMAC, and the evaluation program is called EVAL II and is in FORTRAN. FORMAC is an algebraic manipulation language. FORMAC stands for FORmula MAnipulation Compiler. It is an experimental language developed by IBM, and it runs on an IBM 7094/7044 computer. Bond describes FORMAC in reference 3. There are other algebraic languages which might be used in block diagram reduction. These languages are detailed by Sammet in reference 6. Two programs are necessary to maximize block-diagram-size capability. This approach separates the numerical evaluation from the reduction process. This separation is made since the numerical evaluations are performed more efficiently in FORTRAN than in FORMAC. The two-program approach also gives the user the ability to change component descriptions in EVAL II.

The program listings and flow charts are given in appendix D.

REDUCTION PROGRAM - REDUCE II

REDUCE II is the main program in the block diagram reduction. Initially, the block diagram expressions, the inputs and the output, are read and counted. The reduction order string is examined for algebraic compatibility. That is,

Number of variables in order string = Number of expressions - 1

Since the user may solve for the ratio of an output to an input or an output to all the inputs (see User's Manual (appendix E) for details), the block diagram inputs not used in a specified transfer function are set to zero.

The reduction proceeds by choosing a variable to eliminate, solving for that variable in one of the equations, and substituting the solution for that variable in the remaining equations. Subroutine INSTRN finds the variables in a reduction substring and puts their subscripts in common block STRING. Subroutine SEARCH picks the equations for elimination and substitution in the reduction STRING. SEARCH obtains the variable to be eliminated from the information contained in common block STRING. The general rule is to take the simplest equation as the one to be eliminated. By simplest is meant the equation with the fewest variables. If the variable due for elimination is in more than two equations, some more selective rules are employed. These rules are embodied in

the following priority schedule, which has as its basis identification of the equations in the loop specified by the order substring (when possible):

(1) If another variable in the reduction substring is contained in two or more equations that contain the variable to be eliminated, pick the simplest equation of this new subset of equations to substitute into the other equations that contain the variable to be eliminated.

(2) If there are no more variables in the reduction substring or if rule 1 is not satisfied, look for other variables in common among the equations that contain the variable to be eliminated. If two or more equations contain a variable in common (other than the variable to be eliminated), pick the simplest of these as the equation to be eliminated. If neither rule 1 nor rule 2 can be followed, the simplest equation is picked as the one to be eliminated. Further, if only one (or no) equation contains the variable to be eliminated, an error message is written out and execution is stopped. With this method, the number of equations and variables is reduced by one at each step. The variables that appear in the transfer function are, of course, never eliminated. The final equation contains only the variables of the transfer function and the G-functions.

In the reduction process, new combinations of G's (functions) are formed in subroutine SUPERG. It should be noted that the program assumes that an expression is an equation. That is, if G(1)*X(1)+X(2) is the FORMAC expression, then $G_1X_1 + X_2 = 0$ is the implied equation. An expression prior to super G substitution might look like

$$(1.+G(1)*G(2))*X(1)+G(3)*G(4)*X(5)-G(5)*X(6)$$

The SUPERG subroutine would pick (1.+G(1)*G(2)) as a denominator and construct new coefficients such as G(6) and G(7), where

$$G(6)=G(3)*G(4)/(1.+G(1)*G(2))$$

and

$$G(7)=-G(5)/(1.+G(1)*G(2))$$

SUPERG would punch these new coefficients in FORTRAN in the following style:

```
DENOM(1)=1.+G(1)*G(2)

G(6)=G(3)*G(4)

G(6)=G(6)/DENOM(1)

G(7)=-G(5)

G(7)=G(7)/DENOM(1)
```

Upon leaving subroutine SUPERG this expression would be X(1)+G(6)*X(5)+G(7)*X(6).

The user has the option of selecting one of two forms for a super G. It is possible to always have a denominator of the form

$$1.+G_a*G_b*G_c*G_d+G_e*G_f*G_g+G_k*G_l*G_m\cdots$$

This can be done by picking a coefficient of some variable and adding 1.0 to it. This new term can be divided into all the coefficients of the variables in an expression. This process tends to generate more super G's than when the super G's are allowed to form naturally. By punching the word ARTIFICIAL on the first data card to REDUCE II this form of super G will be produced whenever SUPERG finds a coefficient with two or more terms. If the user does not specify ARTIFICIAL or NATURAL (1.0 is not added to coefficients), an error message is printed out and execution stops.

The output of the REDUCE II program is punched on cards and printed on paper. The cards plus the user's data supply all the information for the EVAL II program. See the user's manual for further information of card input and output. Since the program should be as easy to use and as reliable as possible, the program input is given in algebraic terms for the most part. The user does not have to give a count of how many variables, expressions, inputs, and functions are to follow, as was required in the predecessor of this program. The programs interpret the user's data for this information. The only nonalgebraic piece of data is the first card, which specifies what form the user desires the super G's to have. An attempt has been made to make the programs efficient in terms of execution time. The subroutine COUNTX was written to be faster than the corresponding FORMAC routines that could have been used. Auxiliary array sizes have been kept as small as possible to maximize block-diagram-size capability. It is possible to have a block diagram that is so large that it would be stored on an auxiliary storage device (tape or disk file). If this does happen, there will be a substantial increase in execution time because moving information to and from an auxiliary device is inherently slow. REDUCE II does not call on SIN, COS, TANH, ATAN, ALOG, DUMP, PDUMP, FMCDIF, EXPDMP, or FMCDMP. Using the dummy subroutines SIN, COS, TANH, ATAN, ALOG, DUMP, PDUMP, FMCDIF, EXPDMP, and FMCDMP allows us to use the storage space normally assigned to these routines.

The user may, for any particular job, change any of the dimensions of the program. In particular, if more inputs are desired, the array INPUT should be lengthened. If more equations or variables are needed, the arrays EQ and X should be changed. Also the value of MAX(1) must be changed. If some other number of functions (G's) is to be used, the G array and MAX(2) should be changed accordingly.

EVALUATION PROGRAM - EVAL II

The EVAL II program evaluates the cards REDUCE II punched out. The user inserts some of these cards directly into EVAL II as FORTRAN coding. The remainder is numerical data to identify transfer functions. The program covers a frequency band of FSTART to FEND in steps of DELTAF. The output of EVAL II consists of a tabulation of the transfer functions for each frequency point. Also a set of CALCOMP plots for the frequency band can be obtained. The written information gives a transfer function number, the frequency point, the real and imaginary parts of the transfer function, the absolute value and phase angle of the transfer function, and the magnitude of the transfer function in decibels, based on $dB = 20 \log_{10}$ (absolute value). The plots are of magnitude against frequency and phase angle against frequency. These plots can be made on a linear or semilogarithmic set of axes. The user's manual (appendix E) gives more information on the actual use of EVAL II.

SUBPROGRAMS

REDUCE II uses the following subprograms that are not supplied with either FORTRAN, FORMAC, or this report. These subprograms must be supplied by the user:

Function	Description
ALS(N, X)	Accumulator left shift of X by N binary places
IARS(N, X)	Accumulator right shift of X by N binary places
LGR(N, X)	Logical right shift of X by N binary places
AND(X1, X2)	Logical intersection of X1 and X2
OR(X1, X2)	Logical union of X1 and X2

EVAL II uses the following subprograms that are part of the CALCOMP plotting package. These programs must be user supplied since CALCOMP has proprietary rights on their use. They are LINE, SCALE, AXIS, SYMBOL, NUMBER, and PLOT.

APPENDIX D

DESCRIPTION, FLOW CHARTS, AND LISTINGS FOR COMPUTER PROGRAMS

DESCRIPTION

Reduction Program - REDUCE II

The main program REDUCE II reads the input data, interprets data, and does the actual block diagram reduction. It prints and punches out the transfer function generated.

Subroutine COUNT counts the number of variables in an expression. It returns the number of variables and/or the variables to the calling program.

Subroutine SEARCH finds expressions containing the variable to be eliminated. It constructs a list of the variable to be eliminated, the expression to be eliminated, and the equations to be substituted into.

Subroutine SUPERG creates the super G's or combinations of functions formed in the reduction process.

Subroutine SOUT outputs FORMAC expressions. This output consists of a printed statement, a punched card, or both.

Subroutine INSTRN reads reduction substrings and provides a count of the number of variables and the variables in the substring.

DELETE, REMOVE, FC, and NBR are used to remove $ from FORMAC expressions and to convert floating-point exponents to fixed-point exponents.

Subroutine COS with entry points SIN, TANH, ATAN, ALOG, DUMP, PDUMP, FMCDIF, EXPDMP, and FMCDMP is a dummy subroutine that is not called. It provides entry-point names that are the same as subprograms referenced by FORMAC but are not executed by the programs of this report. This prevents the equivalent subprograms from being loaded and conserves storage space.

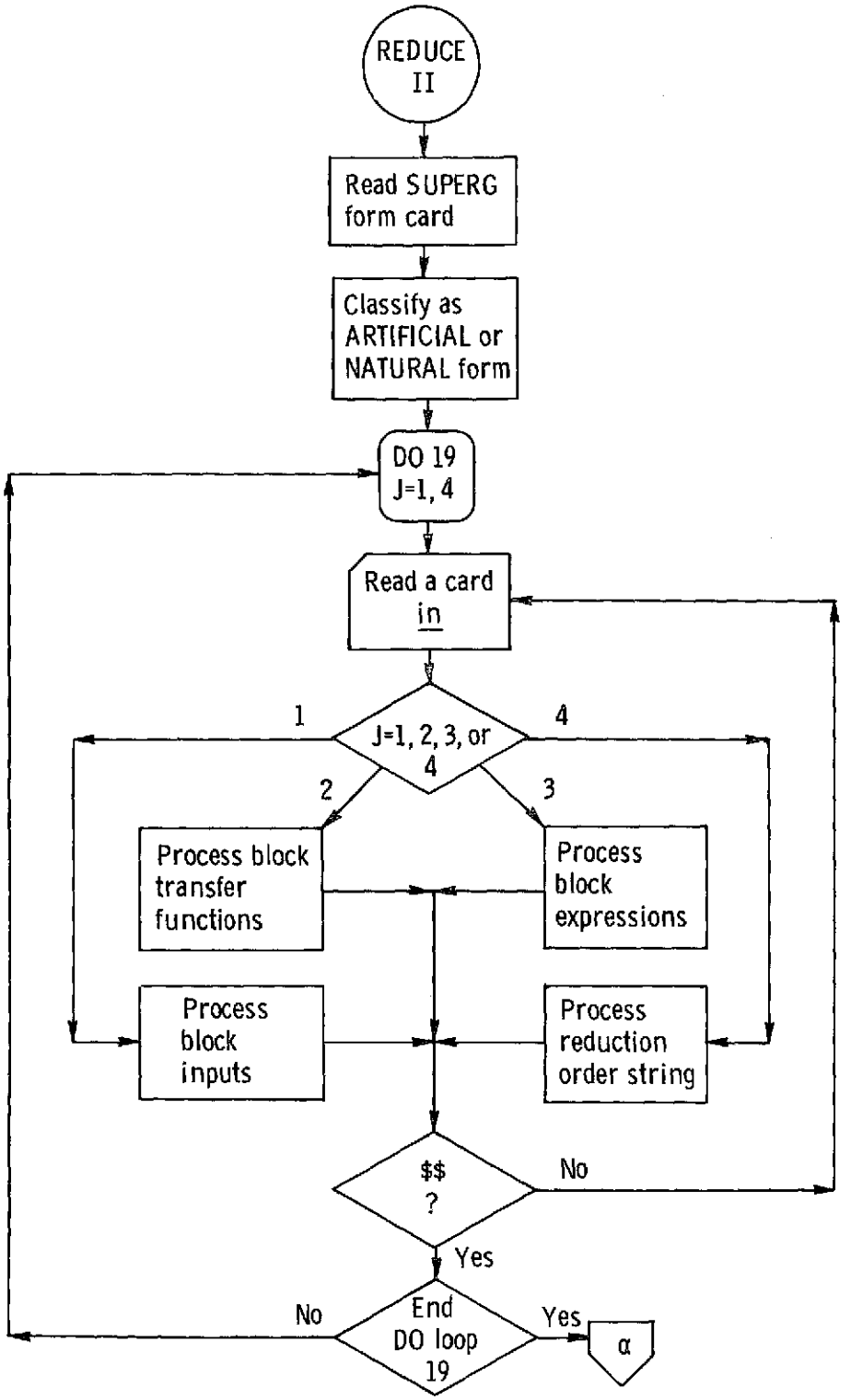
Evaluation Program - EVAL II

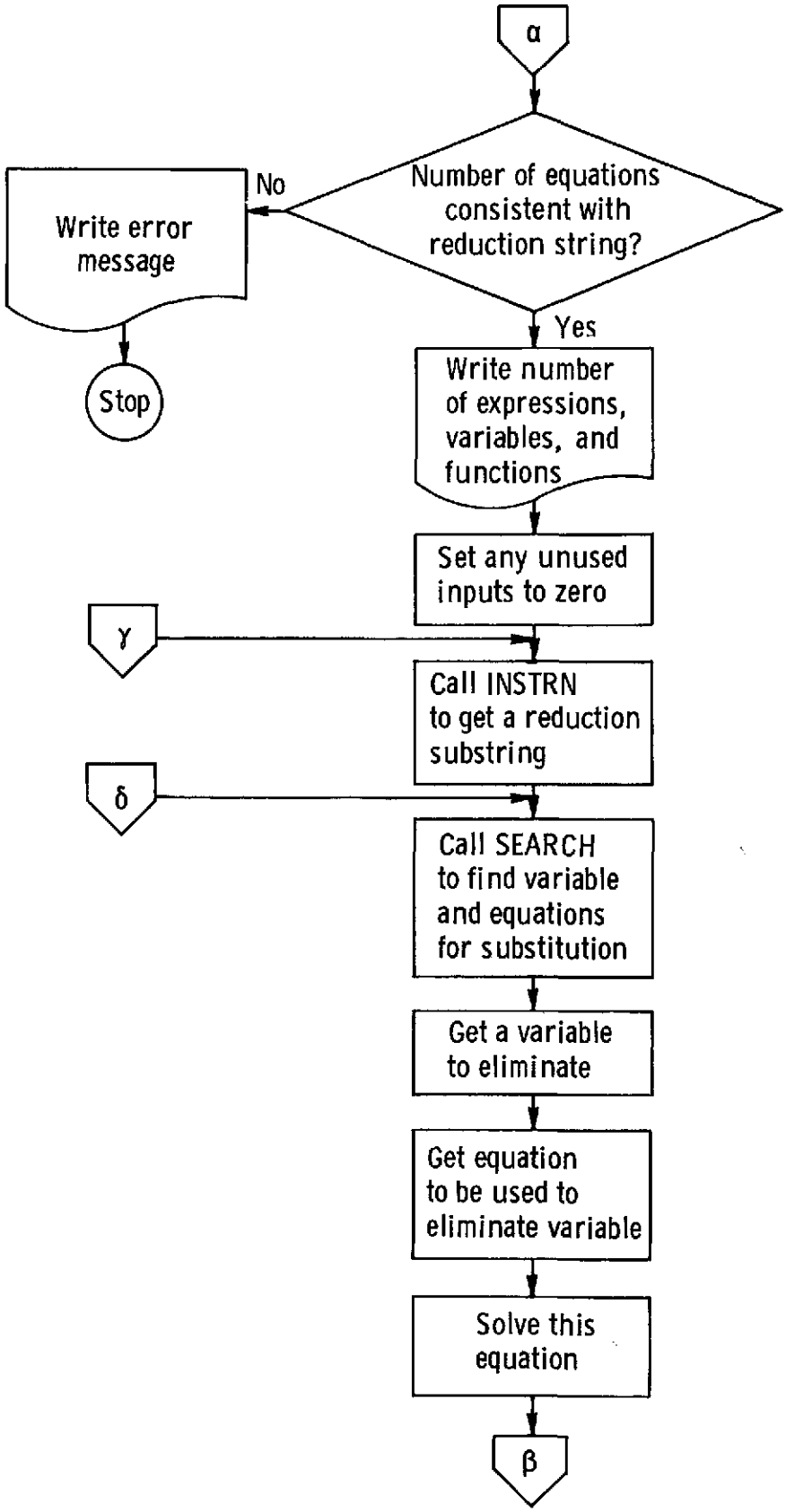
The main program EVAL II reads and writes all the numerical input and output. It calls the subprograms that evaluate the super G's and the transfer functions.

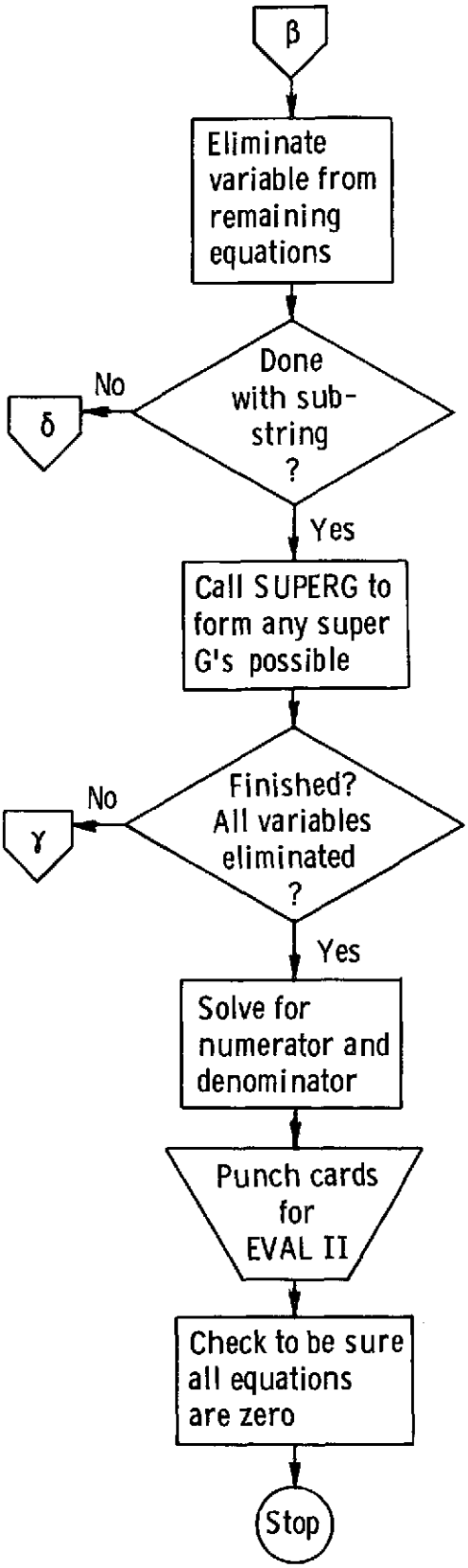
The function subprogram EXPFX, SINFX, COSFX, SINHFX, COSHFX, SQRTFX evaluates the supplied system functions.

The G's, super G's, and transfer function cards (NO=, DO=) are inserted in subroutine COEFF. The actual evaluation of the transfer function occurs in COEFF.

Reduction Program - REDUCE II




α
Number of equations consistent with reduction string?
No
Write error message
Stop
Yes
Write number of expressions, variables, and functions
Set any unused inputs to zero
γ
Call INSTRN to get a reduction substring
δ
Call SEARCH to find variable and equations for substitution
Get a variable to eliminate
Get equation to be used to eliminate variable
Solve this equation
β


β
Eliminate variable from remaining equations
Done with sub-string ?
No
δ
Yes
Call SUPERG to form any super G's possible
Finished? All variables eliminated ?
No
γ
Yes
Solve for numerator and denominator
Punch cards for EVAL II
Check to be sure all equations are zero
Stop

```
      SYMARG                                                                A   1
      ATOMIC X(601),G(1202)                                                 A   2
      COMMON /TRANS/ ARTIF
      DIMENSION IN(70), OUT(17), NEQ(100), EQ(601), INPUT(5), NXARRY(30)    A   3
     1, MINXG(2), GORX(2), MAX(2), LSAVE(50)                                A   4
      DIMENSION IARTIF(3),NATUR(3)
      LOGICAL LOCATE,TEST,FINISH,DONE,ARTIF                                 A   5
      INTEGER OUT
      REAL IS,IT,NUM                                                        A   6
      EQUIVALENCE (IN(1),OUT(1),NEQ(1)), (MINXG(1),M), (MINXG(2),MG)        A   7
      EQUIVALENCE (PLUS1,ARTIF)
      DATA IARTIF(1),NATUR(1)/15HARTIFICIAL FORM,13HNATURAL FORM /
      DATA MAX/600,1200/                                                    A   8
      DATA DR/6H$00000/                                                     A   9
      DATA XX/6HX00000/                                                     A  10
      DATA MINXG,GORX/2*0,1HX,1HG/                                          A  11
C*************************************************************************  A  12
C     M IS THE NUMBER OF VARIABLES                                          A  13
C     M1 IS THE NUMBER OF EQUATIONS                                         A  14
C     IK IS THE NUMBER OF INPUTS                                            A  15
C     MG IS THE NUMBER OF FUNCTIONS AT INPUT                                A  15
C*************************************************************************  A  17
1     CONTINUE                                                              A  21
      PLUS1=3.
      READ(5,42) (IN(I),I=1,3)
      DO 100 I=1,3
      IF(IN(I).EQ.IARTIF(I)) PLUS1=PLUS1+1.
      IF(IN(I).EQ.NATUR(I)) PLUS1=PLUS1-1.
100   CONTINUE
      IF(PLUS1.NE.0..AND.PLUS1.NE.6.) GO TO 380
C*************************************************************************  A  18
C     READ THE BLOCK DIAGRAM PARTS                                          A  19
C*************************************************************************  A  20
      NX=0                                                                  A  22
      M1=1                                                                  A  23
      IK=0                                                                  A  24
      MG=0                                                                  A 24A
      M=0                                                                   A 24B
      DO 19 J=1,4                                                           A  25
      NN=0                                                                  A  26
      PREV=BL                                                               A  27
2     K=1                                                                   A  28
3     K1=14*(K-1)+1                                                         A  29
      K2=K1+13                                                              A  30
      READ (5,42) (IN(L),L=K1,K2)                                           A  31
      GO TO (4,9,13,17),J                                                   A  32
C*************************************************************************  A  33
C     INTERPRET THE BLOCK DIAGRAM INPUTS                                    A  34
C*************************************************************************  A  35
4     CONTINUE                                                              A  36
      DO 5 I=1,80                                                           A  37
      F=FC(IN(K1),I)                                                        A  38
      IF (F.EQ.DR) IK=IK+1                                                  A  39
      IF (F.EQ.DR.AND.PREV.EQ.DR) GO TO 6                                   A  40
      PREV=F                                                                A  41
5     CONTINUE                                                              A  42
      K=K+1                                                                 A  43
      GO TO 3                                                               A  44
```

```
6       JJ=0                                                                    A  45
        IK=IK-1                                                                 A  46
        IF (IK.GT.5) GO TO 35                                                   A  47
        DO 7 JJJ=1,IK                                                           A  48
        LET INPUT(JJJ)=ALGCON IN(1),JJ                                          A  49
7       CONTINUE                                                                A  50
        WRITE (6,54)                                                            A  51
        DO 8 L=1,IK                                                             A  52
        Q=0.0                                                                   A  53
        LET Q = BCDCON INPUT(L),OUT,17                                          A  54
8       WRITE (6,51) OUT(2)                                                     A  55
        GO TO 19                                                                A  56
C*****************************************************************************A  57
C       INTERPRET THE DESIRED BLOCK DIAGRAM OUTPUT                              A  58
C*****************************************************************************A  59
9       CONTINUE                                                                A  60
        JQ=0                                                                    A  61
        DO 10 I=1,80                                                            A  62
10      IF (FC(IN(K1),I).EQ.DR) JQ=JQ+1                                         A  63
        JJ=0                                                                    A  64
        LET NUM = ALGCON IN(1),JJ                                               A  65
        IF (JQ.EQ.2) GO TO 11                                                   A  66
        LET DEN = ALGCON IN(1),JJ                                               A  67
11      CONTINUE                                                                A  68
        Q=0.0                                                                   A  69
        LET Q = BCDCON NUM,OUT,17                                               A  70
        IF (JQ.EQ.3) GO TO 12                                                   A  71
        WRITE (6,46) OUT(2)                                                     A  72
        WRITE (6,49)                                                            A  73
        GO TO 19                                                                A  74
12      CONTINUE                                                                A  75
        WRITE (6,47) OUT(2)                                                     A  76
        Q=0.0                                                                   A  77
        LET Q = BCDCON DEN,OUT,17                                               A  78
        WRITE (6,48) OUT(2)                                                     A  79
        WRITE (6,49)                                                            A  80
        GO TO 19                                                                A  81
C*****************************************************************************A  82
C       INTERPRET THE BLOCK DIAGRAM EXPRESSIONS                                 A  83
C*****************************************************************************A  84
13      CONTINUE                                                                A  85
        Q=1.                                                                    A  86
        DONE=.FALSE.                                                            A  87
        DO 14 I=1,80                                                            A  88
        F=FC(IN(K1),I)                                                          A  89
        IF (F.EQ.DR) Q=0.                                                       A  90
        IF (F.EQ.DR.AND.PREV.EQ.DR) DONE=.TRUE.                                  A 90
        PREV=F                                                                  A  91
14      CONTINUE                                                                A  92
        WRITE (6,53) M1,(IN(KK),KK=K1,K2)                                       A  93
        K=K+1                                                                   A  94
        IF (K.GT.5) GO TO 38                                                    A  95
        IF (Q.NE.0.) GO TO 3                                                    A  96
        DO 16 MM=1,2                                                            A  97
        CALL COUNT (IN(1),GORX(MM),NXS,NXARRY,0)                                A  98
        IF (NXS.EQ.0) GO TO 16                                                  A  99
        DO 15 NN=1,NXS                                                          A 100
15      MINXG(MM)=MAXO(MINXG(MM),NXARRY(NN))                                    A 101
        IF (MINXG(MM).LE.MAX(MM)) GO TO 16                                      A 102
        GO TO (33,32),MM                                                        A 103
```

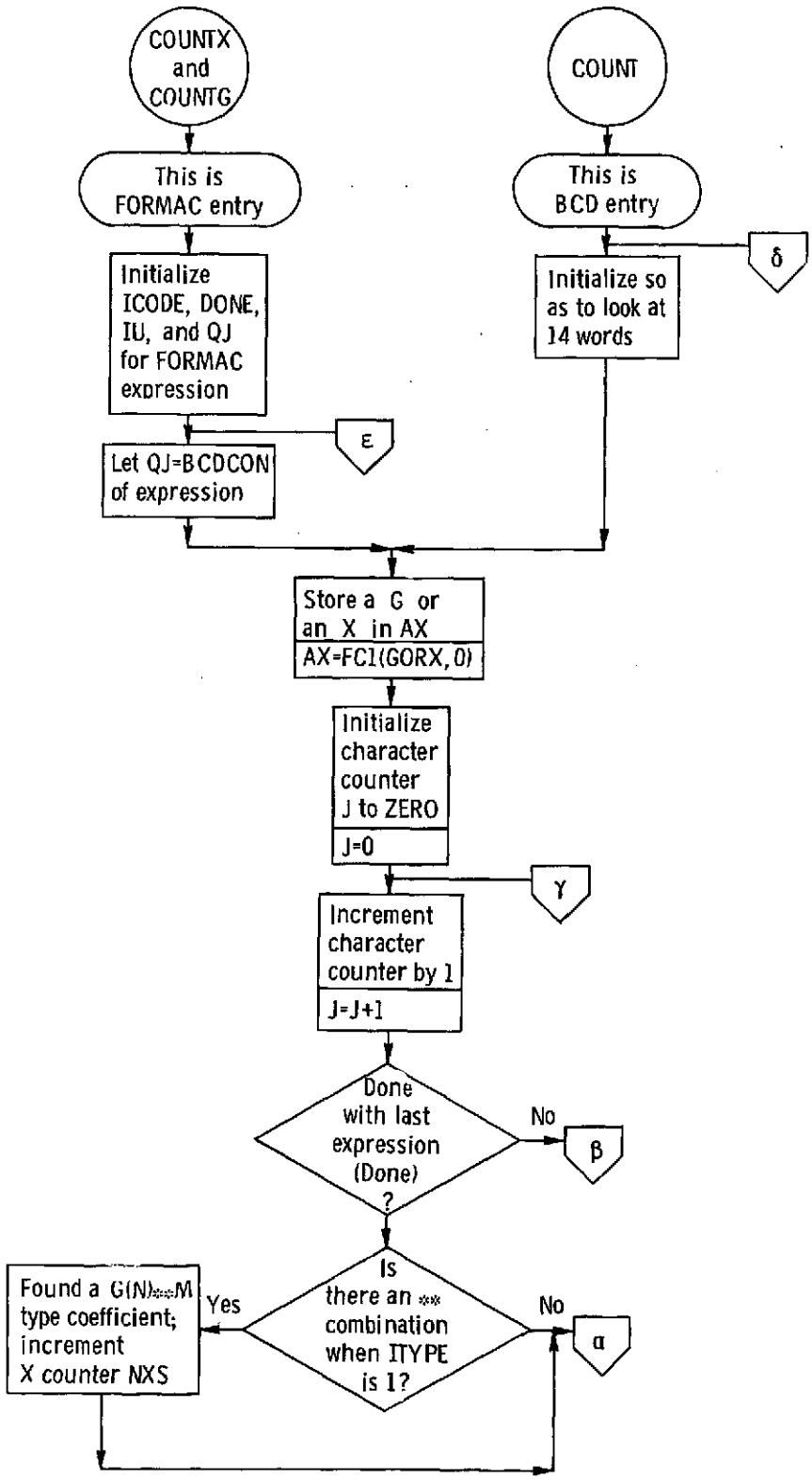
```
16    CONTINUE                                                          A 104
      IF (M1.GT.MAX(1)) GO TO 34                                        A 105
      JJ=0                                                              A 106
      LET EQ(M1)=ALGCON IN(1),JJ                                        A 107
      LET EQ(M1)=EXPAND EQ(M1)                                          A 108
      M1=M1+1                                                           A 109
      IF (.NOT.DONE) GO TO 2                                            A 110
      M1=M1-1                                                           A 111
      GO TO 19                                                          A 112
C*****************************************************************     A 113
C     INTERPRET AND WRITE THE BLOCK REDUCTION ORDER                     A 114
C*****************************************************************     A 115
17    NN=NN+1                                                           A 116
      IF (NN.EQ.1) WRITE (6,44)                                         A 117
      WRITE (6,53) NN,(IN(L),L=K1,K2)                                   A 118
      DO 18 I=1,80                                                      A 119
      CHAR=FC(IN(K1),I)                                                 A 120
      IF (CHAR.EQ.XX) NX=NX+1                                           A 121
18    IF (CHAR.EQ.DR) GO TO 19                                          A 122
      GO TO 2                                                           A 123
19    CONTINUE                                                          A 124
      IF (NX.NE.M1-1) GO TO 37                                          A 125
C*****************************************************************     A 126
C     BACKSPACE THE INPUT UNIT                                          A 127
C*****************************************************************     A 128
      DO 20 L=1,NN                                                      A 129
20    BACKSPACE 5                                                       A 130
      WRITE (6,45) M,M1,MG                                              A 131
      WRITE (6,39)                                                      A 132
C*****************************************************************     A 133
C     INITIALIZE AND SET UP EQUATIONS FOR SOLUTION                      A 134
C*****************************************************************     A 135
C                                                                       A 136
      IF (JQ.EQ.2) GO TO 23                                             A 137
C     REPLACE THE INPUTS BY ZERO IF NOT DEN OR NUM                      A 138
C                                                                       A 139
      DO 22 L=1,M1                                                      A 140
      DO 21 J=1,IK                                                      A 141
      LET TEST = MATCH ID,NUM,INPUT(J)                                  A 142
      IF (TEST) GO TO 21                                                A 143
      LET TEST = MATCH ID,DEN,INPUT(J)                                  A 144
      IF (TEST) GO TO 21                                                A 145
      LET EQ(L) = SUBST EQ(L),(INPUT(J),0)                              A 146
21    CONTINUE                                                          A 147
22    CONTINUE                                                          A 148
C*****************************************************************     A 149
23    CONTINUE                                                          A 150
C*****************************************************************     A 151
C     REDUCE THE EQUATIONS                                              A 152
C*****************************************************************     A 153
24    CONTINUE                                                          A 154
C*****************************************************************     A 155
C     GET THE REDUCTION STRING                                          A 156
C*****************************************************************     A 157
      CALL INSTRN (FINISH)                                              A 158
      LK=0                                                              A 159
C                                                                       A 160
C     GET A VARIABLE TO ELIMINATE                                       A 161
C                                                                       A 162
25    CONTINUE                                                          A 163
```

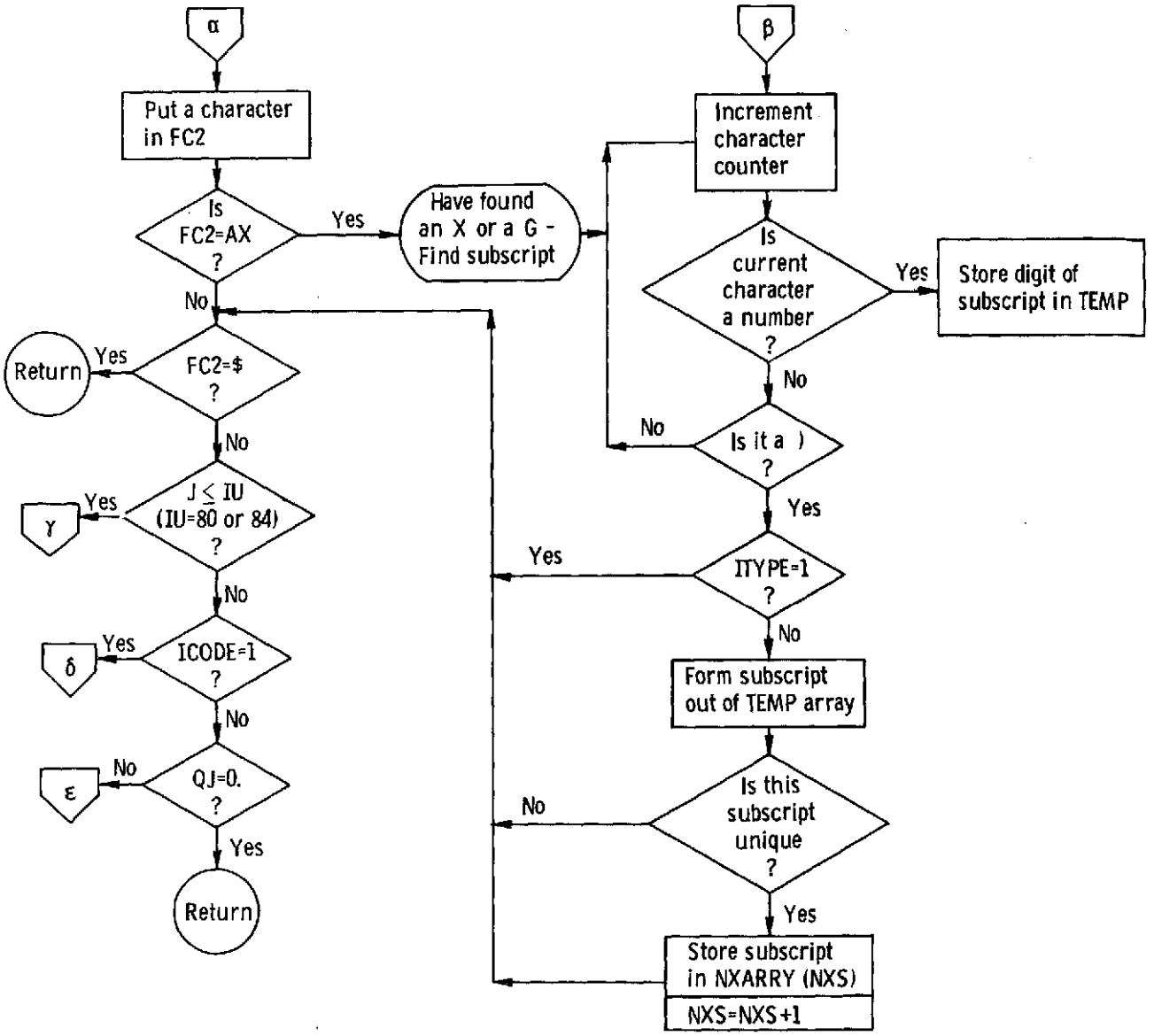
```
      CALL SEARCH (X,EQ,M,M1,NEQ,DONE)                                  A 164
      II=1                                                              A 165
      J=NEQ(II)                                                         A 166
      IF (J.EQ.0) GO TO 27                                              A 167
C                                                                       A 168
C     GET AN EQUATION CONTAINING THE VARIABLE                           A 169
C                                                                       A 170
      II=II+1                                                           A 171
      L=NEQ(II)                                                         A 172
C                                                                       A 173
C     SOLVE THE EQUATION                                                A 174
C                                                                       A 175
      LET RR = EQ(L)                                                    A 176
      LET EQ(L)=0.                                                      A 177
      LET IT = COEFF RR,X(J)                                            A 178
      LET RR = EXPAND RR-IT*X(J)                                        A 179
C                                                                       A 180
C     ELIMINATE THE VARIABLE IN THE REMAINING EQUATIONS                 A 181
C                                                                       A 182
26    II=II+1                                                           A 183
      L=NEQ(II)                                                         A 184
      IF (L.EQ.0) GO TO 27                                              A 185
      LK=LK+1                                                           A 186
      LSAVE(LK)=L                                                       A 187
      LET IS = COEFF EQ(L),X(J)                                         A 188
      LET EQ(L) = EXPAND EQ(L)-IS*X(J)                                  A 189
      LET EQ(L) = EXPAND IS*RR-IT*EQ(L)                                 A 190
      GO TO 26                                                          A 191
27    CONTINUE                                                          A 192
      ERASE IS,IT                                                       A 193
      IF (.NOT.DONE) GO TO 25                                           A 194
      IF(FINISH) GO TO 280
      DO 28 MM=1,LK                                                     A 195
      NN=LK-MM+1                                                        A 196
      L=LSAVE(NN)                                                       A 197
C*************************************************************************A 198
C     CREATE SOME SUPER G'S                                             A 199
C*************************************************************************A 200
      CALL SUPERG (EQ(L),MG,G,X,MAX(2),M)                               A 201
28    CONTINUE                                                          A 202
      GO TO 24
C                                                                       A 204
C                                                                       A 205
280   L=NEQ(II-1)
C*************************************************************************A 207
C     SOLVE FOR NUMERATOR AND DENOMINATOR                               A 208
C*************************************************************************A 209
      LET R = EQ(L)                                                     A 210
      LET RD = COEFF R,NUM                                              A 211
      LET EQ(L)=EQ(L)-RD*NUM                                            A 212
      LET EQ(L)=EXPAND EQ(L)
      LET RD = EXPAND -1*RD                                             A 214
      PUNCH 61                                                          A 215
      CALL SOUT(RD,2,1H )
      CALL COUNTX (NUM,1HX,NXS,NXARRY,0)                                A 217
      NNUM=NXARRY(1)                                                    A 218
      IF (JQ.EQ.3) IK=1                                                 A 220
      DO 30 I=1,IK                                                      A 222
      IF (JQ.EQ.3) GO TO 29                                             A 223
      LET DEN=INPUT(I)                                                  A 224
```

```
29    CONTINUE                                                          A 225
      CALL COUNTX (DEN,1HX,NXS,NXARRY,0)                                A 226
      OUT(I)=NXARRY(1)                                                  A 227
      LET RN = COEFF R,DEN                                              A 229
      LET EQ(L)=EQ(L)-RN*DEN                                            A 230
      LET EQ(L)=EXPAND EQ(L)
C*************************************************************************  A 232
C     WRITE NUMERATOR AND DENOMINATOR                                   A 233
C*************************************************************************  A 234
      WRITE (6,43) NNUM,OUT(I)                                          A 235
      WRITE (6,50)                                                      A 236
      PUNCH 60, I                                                       A 237
      CALL SOUT(RN,3,1H )
      WRITE (6,52)                                                      A 239
      CALL SOUT (RD,1,1H )                                              A 240
30    CONTINUE                                                          A 241
      CALL COMOUT                                                       A 242
      PUNCH 41, NNUM,IK,(OUT(I),I=1,IK)
      ERASE RN,RD                                                       A 243
C*************************************************************************  A 244
C     CHECK ON THE COMPLETEDNESS OF THE SOLUTION                        A 245
C*************************************************************************  A 246
      LET R=0.                                                          A 247
      DO 31 L=1,M1                                                      A 248
      LET TEST= MATCH ID,EQ(L),R                                        A 249
      IF (TEST) GO TO 31                                                A 250
      WRITE (6,62) L,L                                                  A 251
      CALL SOUT (EQ(L),1,1H )
31    CONTINUE                                                          A 253
      GO TO 1                                                           A 254
C*************************************************************************  A 255
C     WRITE ERROR MESSAGES                                              A 256
C*************************************************************************  A 257
32    CONTINUE                                                          A 258
      WRITE (6,57) MAX(2)                                               A 259
      GO TO 36                                                          A 260
33    WRITE (6,55) MAX(1)                                               A 261
      GO TO 36                                                          A 262
34    WRITE (6,56) MAX(1)                                               A 263
      GO TO 36                                                          A 264
35    WRITE (6,58)                                                      A 265
36    STOP                                                              A 266
37    WRITE (6,40)                                                      A 267
      GO TO 36                                                          A 268
38    CONTINUE                                                          A 269
      WRITE (6,59)                                                      A 270
      GO TO 36                                                          A 271
380   WRITE(6,63)
      STOP
C                                                                       A 272
C                                                                       A 273
39    FORMAT (//10X,33H* * * OUTPUT FROM REDUCE II * * *)               A 274
40    FORMAT (10X,114H**ERROR** NUMBER OF VARIABLES IN REDUCTION STRING  A 275
     1IS NOT CONSISTENT WITH NUMBER OF EXPRESSIONS--EXECUTION STOPPED )  A 276
41    FORMAT (I5 )                                                      A 277
42    FORMAT (13A6,A2)                                                  A 278
43    FORMAT (1HK,10X,20HTRANSFER FUNCTION X(,I3,4H)/X(,I3,2H) )        A 279
44    FORMAT (23HKTHE REDUCTION ORDER IS,//)                            A 280
45    FORMAT (34HOTHE LARGEST VARIABLE SUBSCRIPT IS,I5,/,29H THE NUMBER  A 281
     1OF EXPRESSIONS IS,I5,/,38H THE LARGEST FUNCTION (G) SUBSCRIPT IS,I  A 282
```

```
     24)                                                                    A 283
46    FORMAT (21HKTHE SOLUTION IS FOR ,A6,14H TO THE INPUTS)                A 284
47    FORMAT (21HKTHE SOLUTION IS FOR ,A6,2HTO)                             A 285
48    FORMAT (1H+,29X,A6)                                                   A 286
49    FORMAT (30HKTHE BLOCK DIAGRAM EXPRESSIONS,//)                         A 287
50    FORMAT (1HK,15X,13HTHE NUMERATOR)                                     A 288
51    FORMAT (5X,9A6)                                                       A 289
52    FORMAT (1HK,15X,15HTHE DENOMINATOR)                                   A 290
53    FORMAT (3X,I4,2X,13A6,A2)                                             A 291
54    FORMAT (25H1THE BLOCK DIAGRAM INPUTS,//)                              A 292
55    FORMAT (20H **ERROR** MORE THAN,I4,10H VARIABLES)                     A 293
56    FORMAT (20H **ERROR** MORE THAN,I4,10H EQUATIONS)                     A 294
57    FORMAT (20H **ERROR** MORE THAN,I4,10H FUNCTIONS)                     A 295
58    FORMAT (43H **ERROR** MORE THAN 5 BLOCK DIAGRAM INPUTS)               A 296
59    FORMAT (59H **ERROR** MORE THAN 5 CARDS FOR A BLOCK DIAGRAM EXPRES    A 297
     1SION)                                                                 A 298
60    FORMAT (6X,3HND(,I2,2H)=)                                             A 299
61    FORMAT (6X,3HDO=)                                                     A 300
62    FORMAT (//,10X,69H***** THE SOLUTION FOR THIS PROBLEM MAY BE IN ER    A 301
     1ROR BECAUSE EQUATION ,I4,31H WAS NOT ELIMINATED, EQUATION  ,I4,7HI    A 302
     2S*****)                                                               A 303
63    FORMAT (10X, 40HINVALID OPTION FOR THE SUPER G FORM CARD)
      END                                                                   A 304-
```

Subroutines COUNTX, COUNTG, and COUNT

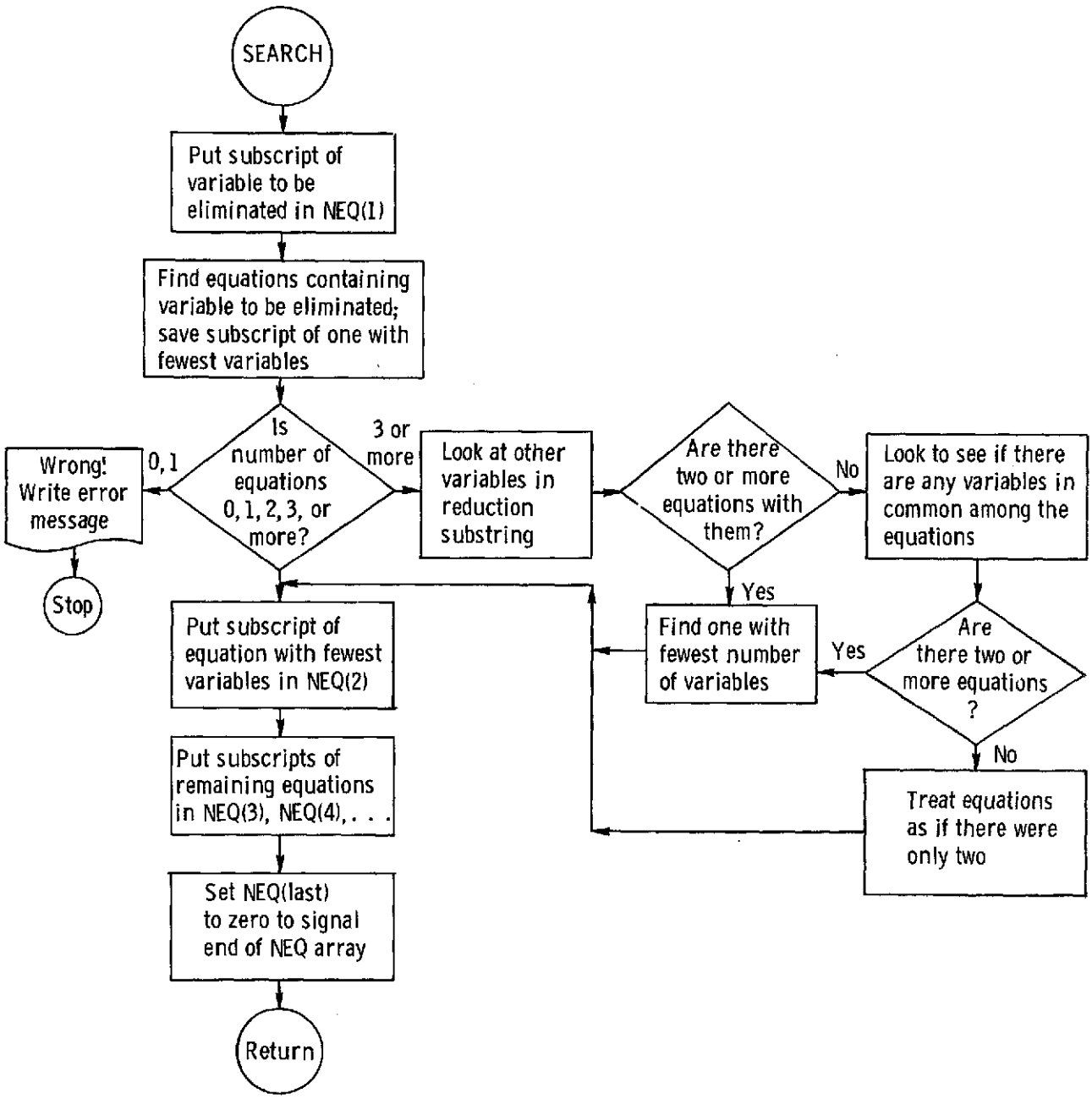



α
Put a character in FC2
Is FC2=AX ?
Yes
Have found an X or a G - Find subscript
No
Return
Yes
FC2=$?
No
Yes
γ
J ≤ IU (IU=80 or 84) ?
No
Yes
δ
ICODE=1 ?
No
No
ε
QJ=0. ?
Yes
Return
β
Increment character counter
Is current character a number ?
Yes
Store digit of subscript in TEMP
No
No
Is it a) ?
Yes
Yes
ITYPE=1 ?
No
Form subscript out of TEMP array
No
Is this subscript unique ?
Yes
Store subscript in NXARRY (NXS)
NXS=NXS+1

```
      SUBROUTINE COUNTX (EQ,GORX,NXS,NXARRY,ITYPE)                      B   1
      ENTRY COUNTG(EQ,NXS,GORX,ITYPE)                                   B   2
      SYMARG EQ                                                         B   3
      DIMENSION NXARRY(1), MTEMP(4), OUT(15), TEMP(4), OTEMP(14)        B   4
      DIMENSION FC2(1)                                                  B   5
      EQUIVALENCE (MTEMP,TEMP)                                          B   6
      EQUIVALENCE (OUT(2),OTEMP(1))                                     B   7
      LOGICAL NBR,DONE                                                  B   8
      LOGICAL ZERO                                                      B   9
      DATA AST,DL,RP/6H*00000,6H$00000,6H)00000/                        B  10
      NXS=0                                                             B  11
      ICODE=0                                                           B  12
      DONE=.TRUE.                                                       B  13
      IU=84                                                             B  14
      QJ=0.                                                             B  15
1     CONTINUE                                                          B  16
      LET QJ=BCDCON EQ,OUT,15                                           B  17
      GO TO 4                                                           B  18
      ENTRY COUNT(ARRAY,GORX,NXS,NXARRY,ITYPE)                          B  19
      DIMENSION ARRAY(1)                                                B  20
      ICODE=1                                                           B  21
      DONE=.TRUE.                                                       B  22
      NXS=0                                                             B  23
      IU=80                                                             B  24
      K=0                                                               B  25
2     K=K+1                                                             B  26
      K1=(K-1)*14+1                                                     B  27
      K2=K1+13                                                          B  28
      L=0                                                               B  29
      DO 3 I=K1,K2                                                      B  30
      L=L+1                                                             B  31
3     OTEMP(L)=ARRAY(I)                                                 B  32
4     CONTINUE                                                          B  33
      IF (.NOT.DONE) GO TO 5                                            B  34
      AX=FC1(GORX,0)                                                    B  35
      PREV1=AX                                                          B  36
5     CONTINUE                                                          B  37
      J=0                                                               B  38
6     J=J+1                                                             B  39
      IF (.NOT.DONE) GO TO 8                                            B  40
      FC2=FC(OTEMP,J)                                                   B  41
      IF (ITYPE.NE.1) GO TO 7                                           B  42
      IF (PREV1.EQ.AST.AND.FC2.EQ.AST) NXS=NXS+1                        B  43
      PREV1=FC2                                                         B  44
7     CONTINUE                                                          B  45
      IF (FC2.NE.AX) GO TO 14                                           B  46
      DONE=.FALSE.                                                      B  47
      NXS=NXS+1                                                         B  48
      KTR=0                                                             B  49
      J=J+2                                                             B  50
      IF (J.GT.IU) GO TO 15                                             B  51
8     CONTINUE                                                          B  52
      DO 10 J2=J,IU                                                     B  53
      J3=J2                                                             B  54
      FC2=FC(OTEMP,J2)                                                  B  55
      IF (FC2.EQ.RP) GO TO 9                                            B  56
      IF (.NOT.NBR(FC2,1)) GO TO 10                                     B  57
      KTR=KTR+1                                                         B  58
```

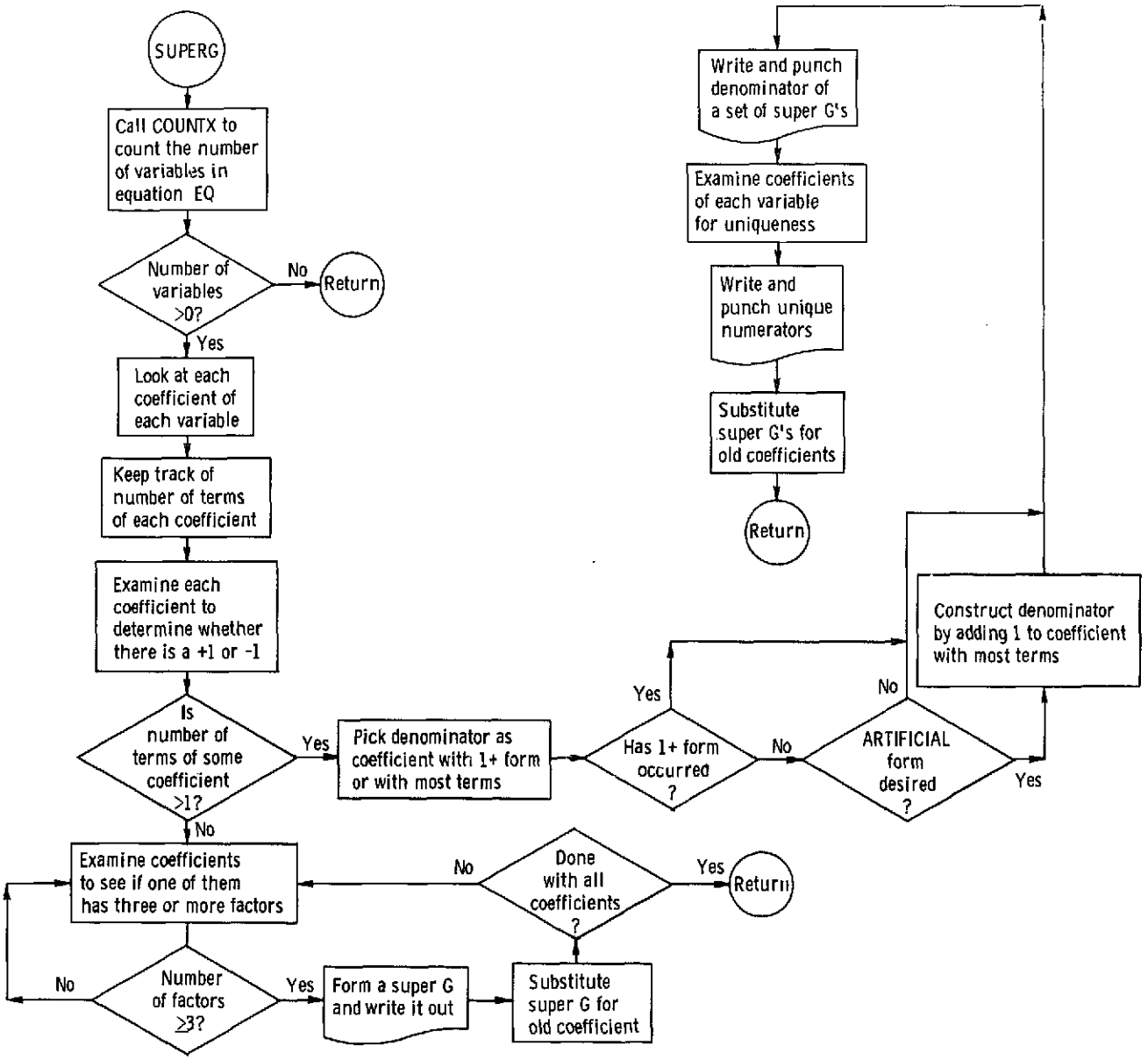
```
      TEMP(KTR)=ARS(30,FC2)                                              B  59
      IF (J2.EQ.IU) GO TO 15                                             B  60
      GO TO 10                                                           B  61
 9    CONTINUE                                                           B  62
      DONE=.TRUE.                                                        B  63
      GO TO 11                                                           B  64
 10   CONTINUE                                                           B  65
 11   CONTINUE                                                           B  66
      J=J3                                                               B  67
      IF (.NOT.DONE) GO TO 15                                            B  68
      IF (ITYPE.EQ.1) GO TO 14                                           B  69
      NXARRY(NXS)=0                                                      B  70
      ITEN=1                                                             B  71
      DO 12 II=1,KTR                                                     B  72
      KTR1=KTR-II+1                                                      B  73
      NXARRY(NXS)=ITEN*MTEMP(KTR1)+NXARRY(NXS)                           B  74
 12   ITEN=ITEN*10                                                       B  75
      IF (NXS.EQ.1) GO TO 14                                             B  76
      NXS1=NXS-1                                                         B  77
      DO 13 II=1,NXS1                                                    B  78
      IF (NXARRY(NXS).NE.NXARRY(II)) GO TO 13                            B  79
      NXS=NXS-1                                                          B  80
      GO TO 14                                                           B  81
 13   CONTINUE                                                           B  82
 14   CONTINUE                                                           B  83
      DONE=.TRUE.                                                        B  84
      IF (FC2.EQ.DL) GO TO 16                                            B  85
      IF (J.LT.IU) GO TO 6                                               B  86
 15   CONTINUE                                                           B  87
      IF (ICODE.EQ.1) GO TO 2                                            B  88
      IF (QJ.NE.0.) GO TO 1                                              B  89
 16   CONTINUE                                                           B  90
      RETURN                                                             B  91
      END                                                                B  92-
```

Subroutine SEARCH



```
      SUBROUTINE SEARCH (X,EQ,MX,MEQ,NEQ,DONE)                          C    1
      DIMENSION NEQ(1), NARRAY(30), EQ(1)                               C    2
      COMMON /STRING/ IA(200),M                                         C    3
      SYMARG X,EQ                                                       C    4
      ATOMIC X(1)                                                       C    5
      LOGICAL DONE,Q                                                    C    6
      DATA II/0/                                                        C    7
      DONE=.FALSE.                                                      C    8
      II=II+1                                                           C    9
      NEQ(1)=IA(II)                                                     C   10
      L=0                                                               C   11
      MIN=999                                                           C   12
      DO 2 I=1,MEQ                                                      C   13
      CALL COUNTX (EQ(I),1HX,NXS,NARRAY,0)                              C   14
      IF (NXS.EQ.0) GO TO 2                                             C   15
      DO 1 J=1,NXS                                                      C   16
      K=NARRAY(J)                                                       C   17
      IF (K.NE.NEQ(1)) GO TO 1                                          C   18
      L=L+1                                                             C   19
      NEQ(L+2)=I                                                        C   20
      IF (NXS.GE.MIN) GO TO 2                                           C   21
      MIN=NXS                                                           C   22
      IMIN=I                                                            C   23
      GO TO 2                                                           C   24
      CONTINUE                                                          C   25
      CONTINUE                                                          C   26
      IMSAVE=IMIN                                                       C   27
      IF (L-2) 3,4,9                                                    C   28
      ERROR  ONLY ONE EQUATION WITH VAR X(NEQ(1))                       C   29
      WRITE (6,17) NEQ(1)                                               C   30
      STOP                                                              C   31
      NEQ(2)=IMIN                                                       C   32
      L1=L+1                                                            C   33
      L2=L+2                                                            C   34
      DO 6 I=3,L1                                                       C   35
      IF (NEQ(I).NE.IMIN) GO TO 6                                       C   36
      I1=I+1                                                            C   37
      DO 5 J=I1,L2                                                      C   38
      NEQ(J-1)=NEQ(J)                                                   C   39
      GO TO 7                                                           C   40
      CONTINUE                                                          C   41
      CONTINUE                                                          C   42
      NEQ(L2)=0                                                         C   43
      IF (II.NE.M) GO TO 8                                              C   44
      II=0                                                              C   45
      DONE=.TRUE.                                                       C   46
      CONTINUE                                                          C   47
      RETURN                                                            C   48
      IF (II.EQ.M) GO TO 13                                             C   49
      I1=II+1                                                           C   50
      DO 12 I=I1,M                                                      C   51
      IS=0                                                              C   52
      MIN=999                                                           C   53
      DO 11 J=1,L                                                       C   54
      K=NEQ(J+2)                                                        C   55
      CALL COUNTX (EQ(K),1HX,NXS,NARRAY,0)                              C   56
      DO 10 N=1,NXS                                                     C   57
      IF (NARRAY(N).NE.IA(I)) GO TO 10                                  C   58
```

```
      IS=IS+1                                                          C  59
      IF (NXS.GE.MIN) GO TO 11                                         C  60
      MIN=NXS                                                          C  61
      IMIN=K                                                           C  62
      GO TO 11                                                         C  63
10    CONTINUE                                                         C  64
11    CONTINUE                                                         C  65
      IF (IS.LT.2) GO TO 12                                            C  66
      GO TO 4                                                          C  67
12    CONTINUE                                                         C  68
13    L1=L-1                                                           C  69
      DO 16 I=1,L1                                                     C  70
      K=NEQ(I+2)                                                       C  71
      CALL COUNTX (EQ(K),1HX,NXS,NARRAY,0)                             C  72
      J=I+1                                                            C  73
      DO 15 N=J,L                                                      C  74
      K1=NEQ(N+2)                                                      C  75
      DO 14 N1=1,NXS                                                   C  76
      N2=NARRAY(N1)                                                    C  77
      IF (N2.EQ.NEQ(1)) GO TO 14                                       C  78
      LET Q= FIND EQ(K1),APP,ALL,(X(N2))                               C  79
      IF (.NOT.Q) GO TO 14                                             C  80
      CALL COUNTG (EQ(K1),NX1,1HX,1)                                   C  81
      IMIN=K                                                           C  82
      IF (NX1.LT.NXS) IMIN=K1                                          C  83
      GO TO 4                                                          C  84
14    CONTINUE                                                         C  85
15    CONTINUE                                                         C  86
16    CONTINUE                                                         C  87
      IMIN=IMSAVE                                                      C  88
      GO TO 4                                                          C  89
C                                                                      C  90
17    FORMAT (10X,36H**** ONLY ONE EQUATION CONTAINING X(,I4,6H) ****) C  91
      END                                                              C  92-
```

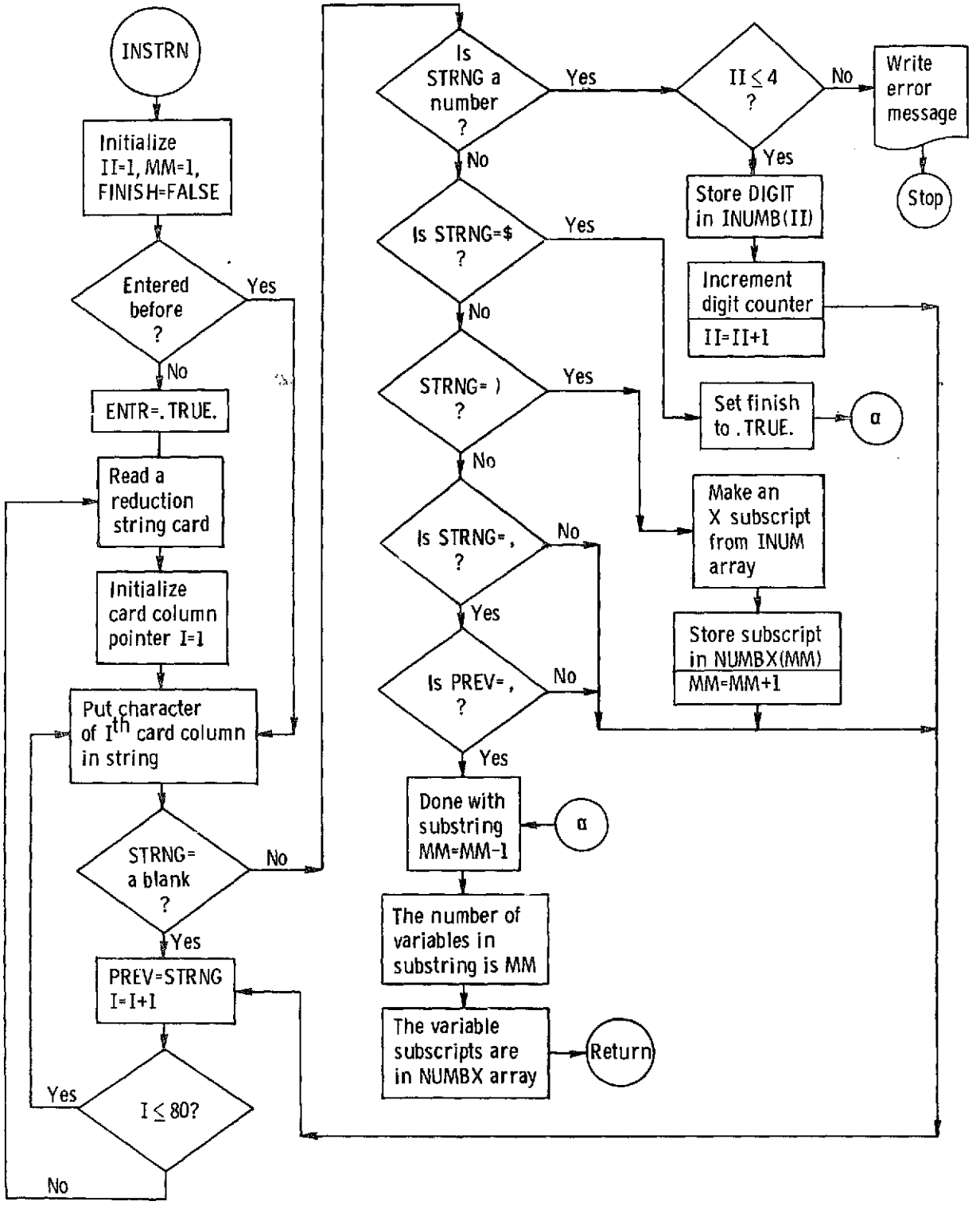

SUPERG
Call COUNTX to count the number of variables in equation EQ
Number of variables >0?
No
Return
Yes
Look at each coefficient of each variable
Keep track of number of terms of each coefficient
Examine each coefficient to determine whether there is a +1 or -1
Is number of terms of some coefficient >1?
Yes
Pick denominator as coefficient with 1+ form or with most terms
No
Examine coefficients to see if one of them has three or more factors
Number of factors ≥3?
No
Yes
Form a super G and write it out
Substitute super G for old coefficient
Done with all coefficients ?
No
Yes
Return
Write and punch denominator of a set of super G's
Examine coefficients of each variable for uniqueness
Write and punch unique numerators
Substitute super G's for old coefficients
Return
Has 1+ form occurred ?
Yes
No
ARTIFICIAL form desired ?
No
Yes
Construct denominator by adding 1 to coefficient with most terms

```
SUBROUTINE SUPERG (EQ,MG,G,X,MAXG,MX)                              D    1
SYMARG EQ,X,G                                                      D    2
ATOMIC X(1),G(1)                                                   D    3
DIMENSION NARRAY(50)                                               D    4
DIMENSION ISAME(50)                                                D    5
COMMON /TRANS/ ARTIF
LOGICAL TRUTH                                                      D    6
LOGICAL FIRST                                                      D    7
LOGICAL PRESG                                                      D    8
LOGICAL YES                                                        D    9
LOGICAL ARTIF
DATA FIRST/.TRUE./                                                 D   10
DATA MD/0/                                                         D   11
IF (MG.GT.MAXG) RETURN                                             D   12
LET EQSAVE=EQ                                                      D   13
MAX=-1                                                             D   14
COUNT THE NUMBER OF X'S IN EQUATION EQ                             D   15
CALL COUNTX (EQ,1HX,NXS,NARRAY,0)                                  D   16
IF (NXS.EQ.0) GO TO 19                                             D   17
DO 5 K=1,NXS                                                       D   18
I=NARRAY(K)                                                        D   19
LET RR=COEFF EQ,X(I)                                               D   20
LET IQ=CENSUS RR,TERM                                              D   21
LET YES= FIND RR,APP,ALL,( 1.)                                     D   22
IF (YES) GO TO 1                                                   D   23
LET YES= FIND RR,APP,ALL,(-1.)                                     D   24
IF (YES) GO TO 1                                                   D   25
IQ=-IQ                                                             D   25
CONTINUE                                                           D   27
IF (IABS(IQ).EQ.1) GO TO 5                                         D   28
IF (MAX.GT.0.AND.IQ.GT.0) GO TO 2                                  D   29
IF (MAX.LT.0.AND.IQ.GT.0) GO TO 3                                  D   30
IF (MAX.LT.0.AND.IQ.LT.0) GO TO 4                                  D   31
GO TO 5                                                            D   32
IF (IQ.LT.MAX) GO TO 5                                             D   33
MAX=IQ                                                             D   34
L=I                                                                D   35
GO TO 5                                                            D   36
IF (IQ.LT.MAX) GO TO 3                                             D   37
CONTINUE                                                           D   38
IF (MAX.NE.-1) GO TO 7                                             D   39
DO 6 K=1,NXS                                                       D   40
I=NARRAY(K)                                                        D   41
LET RR=COEFF EQ,X(I)                                               D   42
CALL COUNTG (RR,NGS,1HG,1)                                         D   43
IF (NGS.LT.3) GO TO 6                                              D   44
FORM A NEW SUPER G                                                 D   45
MG=MG+1                                                            D   46
IF (MG.GT.MAXG) GO TO 20                                           D   47
L=1                                                                D   48
LET EQ = EQ - RR * X(L)                                            D   49
LET EQ=EXPAND EQ+ G(MG)*X(L)                                       D   50
WRITE (6,24)                                                       D   51
WRITE (6,21) MG                                                    D   52
PUNCH 21, MG
CALL SOUT(RR,3,1H+)
CONTINUE                                                           D   54
GO TO 19                                                           D   55
```

```
C       DIVISION SEARCH AND CONSTRUCTION OF SUPER G'S                    D  56
7       CONTINUE                                                         D  57
        PRESG=.FALSE.                                                    D  58
        LET RR=COEFF EQ,X(L)                                             D  59
        LET EQ=EQ-RR*X(L)                                                D  60
        LET DD=1.                                                        D  61
        IF (.NOT.ARTIF.OR.MAX.GT.0) GO TO 8                              D  62
        LET SS=RR                                                        D  63
        LET RR=RR+1.                                                     D  64
        PRESG=.TRUE.                                                     D  65
8       CONTINUE                                                         D  66
        MD=MD+1                                                          D  67
        WRITE (6,24)                                                     D  68
        PUNCH 22,MD
        WRITE (6,22) MD                                                  D  69
        CALL SOUT(RR,3,1H+)
        IF (.NOT.PRESG) GO TO 9
        MG=MG+1
        IF (MG.GT.MAXG) GO TO 20
        WRITE(6,24)
        WRITE(6,21) MG
        CALL SOUT(SS,1,1H+)
        LET DD=G(MG)
        PUNCH 27,MG,MD,MD
        WRITE(6,23) MG,MG,MD
9       CONTINUE                                                         D  72
        KTR=0                                                            D  73
        PRESG=.FALSE.                                                    D  74
        LET EQ=EXPAND EQ+DD*X(L)                                         D  75
        KTR1=KTR                                                         D  76
        DO 18 I=1,NXS                                                    D  77
        N=NARRAY(I)                                                      D  78
        IF (N.EQ.L) GO TO 18                                             D  79
        IF (KTR.EQ.0) GO TO 11                                           D  80
        DO 10 J=1,KTR                                                    D  81
10      IF (N.EQ.IABS(ISAME(J))) GO TO 18                                D  82
11      KTR1=KTR1+1                                                      D  83
        ISAME(KTR1)=N                                                    D  84
        LET SS=COEFF EQ,X(N)                                             D  85
        DO 15 J=1,NXS                                                    D  86
        K=NARRAY(J)                                                      D  87
        IF (K.EQ.L.OR.K.EQ.N) GO TO 15                                   D  88
        IF (KTR.EQ.0) GO TO 13                                           D  89
        DO 12 KK=1,KTR                                                   D  90
12      IF (K.EQ.IABS(ISAME(KK))) GO TO 15                               D  91
13      CONTINUE                                                         D  92
        LET TT= COEFF EQ,X(K)                                            D  93
        IS=1                                                             D  94
        LET TRUTH=MATCH ID,SS,TT                                         D  95
        IF (TRUTH) GO TO 14                                              D  96
        LET TRUTH= MATCH ID,SS,-TT                                       D  97
        IF (.NOT.TRUTH) GO TO 15                                         D  98
        IS=-1                                                            D  99
14      KTR1=KTR1+1                                                      D 100
        ISAME(KTR1)=K*IS                                                 D 101
15      CONTINUE                                                         D 102
16      CONTINUE                                                         D 103
        MG=MG+1                                                          D 104
        IF (MG.GT.MAXG) GO TO 20                                         D 105
        WRITE (6,24)                                                     D 106
```

```
      WRITE (6,21) MG                                                    D 107
      PUNCH 21,MG
      CALL SOUT(SS,3,1H+)
      WRITE (6,23) MG,MG,MD                                              D 109
      PUNCH 23,MG,MG,MD
      LET DD=G(MG)                                                       D 110
      L1=KTR+1                                                           D 112
      DO 17 J=L1,KTR1                                                    D 113
      M=IABS(ISAME(J))                                                   D 114
      S=ISIGN(1,ISAME(J))                                                D 115
      LET EQ=EQ-S*SS*X(M)                                                D 116
      LET EQ= EXPAND EQ+S*DD*X(M)                                        D 117
17    CONTINUE                                                           D 118
      ERASE DD,TT,SS                                                     D 119
      KTR=KTR1                                                           D 120
      IF (KTR.EQ.NXS) GO TO 19                                           D 121
18    CONTINUE                                                           D 122
19    CONTINUE                                                           D 123
      ERASE RR                                                           D 124
      RETURN                                                             D 125
20    WRITE (6,26)                                                       D 126
      LET EQ=EQSAVE                                                      D 127
      RETURN                                                             D 128
      ENTRY COMOUT                                                       D 129
      MD=MAXO(MD,1)
      PUNCH 25, MG,MD                                                    D 130
      RETURN                                                             D 131
C                                                                        D 132
C                                                                        D 133
21    FORMAT (     7X,2HG(,I4,2H)=)                                      D 134
22    FORMAT (     7X,6HDENOM(,I4,2H)=)
23    FORMAT (     7X,2HG(,I4,4H)=G(,I4,8H)/DENOM(,I4,1H))
24    FORMAT (1HK)                                                       D 137
25    FORMAT (6X,10HCOMPLEX G(,I4,8H),DENOM(,I4,1H))                     D 138
26    FORMAT (105HK**** WARNING THE NUMBER OF SUPER FUNCTIONS (G'S) EXCE D 139
     1EDS THE MAXIMUM NUMBER OF FUNCTIONS (G'S) PERMITTED,/,10X,43HTHERE D 140
     2FORE SUBSTITUTION PROCESS STOPPED ****)                           D 141
27    FORMAT(7X,2HG(,I4,9H)=(DENOM(,I4,12H)-1.)/DENOM(,I4,1H))
      END                                                                D 142-
```

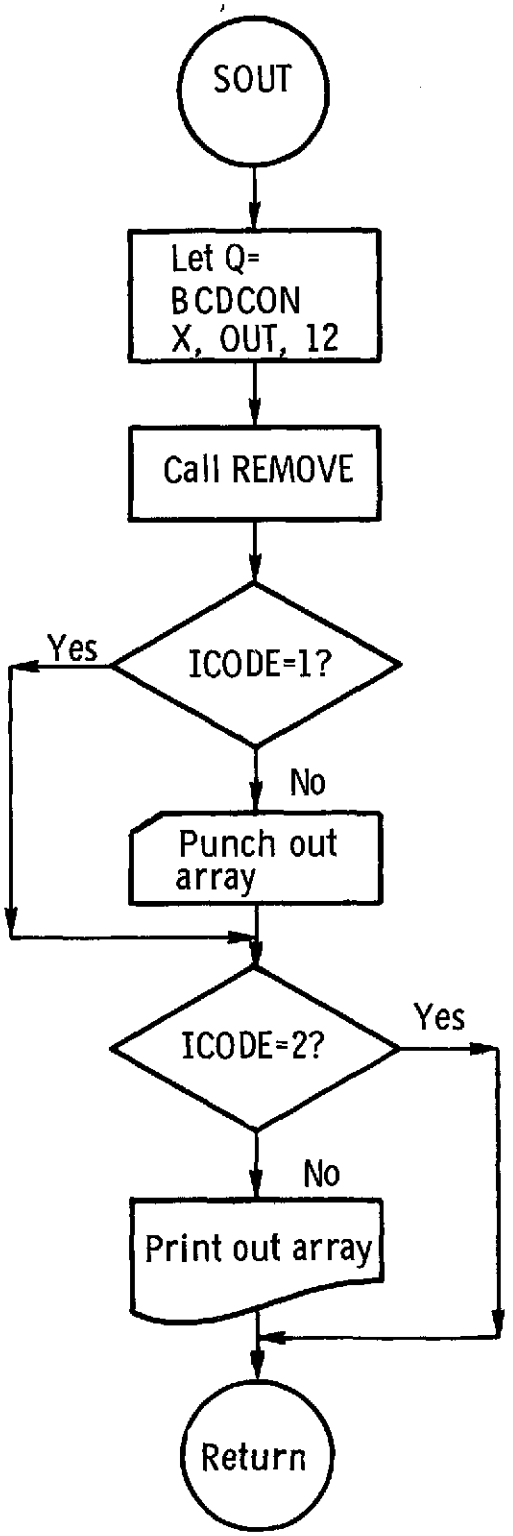
Subroutine INSTRN



```
      SUBROUTINE INSTRN (FINISH)                                        E   1
      COMMON /STRING/ NUMBX(200),MM                                     E   2
      DIMENSION ASTRNG(14), INUM(4)                                     E   3
      LOGICAL ENTR                                                      E   4
      DATA ENTR/.FALSE./                                                E   5
      LOGICAL FINISH                                                    E   8
      DATA BLANK,DOLRSN,RGHTPR,COMMA/1H ,1H$,1H),1H,/                   E   6
      DATA INUM/4*0/                                                    E   7
      DATA MBL/00006060606060/
      FINISH=.FALSE.                                                    E   9
      II=1                                                              E  10
      MM=1                                                              E  11
      IF (ENTR) GO TO 2                                                 E  12
      ENTR=.TRUE.                                                       E  13
1     READ (5,10) (ASTRNG(J),J=1,14)                                    E  14
      I=1                                                               E  15
2     CONTINUE                                                          E  16
      STRNG=OR(FC(ASTRNG,I),MBL)
      IF (STRNG.EQ.BLANK) GO TO 8                                       E  17
C     NOT A BLANK,SEARCH FOR A NUMBER                                   E  18
      NUMBER=LGR(30, STRNG    )                                         E  19
      IF (NUMBER.LT.0.OR.NUMBER.GT.9) GO TO 3                           E  20
C     FOUND ONE OF THE INTEGERS N OF THE X SUBSCRIPT X(NNNN)            E  21
      IF (II.GT.4) GO TO 9                                              E  22
      INUM(II)=NUMBER                                                   E  23
      II=II+1                                                           E  24
      GO TO 8                                                           E  25
3     IF (STRNG.NE.DOLRSN) GO TO 4                                      E  26
C     DONE WITH ALL INPUT STRINGS,TERMINATE BUFFERS AND RETURN          E  27
      FINISH=.TRUE.                                                     E  28
      GO TO 7                                                           E  29
4     IF ( STRNG    .NE.RGHTPR) GO TO 6                                 E  30
C     TERMINATE NUMBER BUFFER AND CONSTRUCT NNNN OF X(NNNN)             E  31
      II=II-1                                                           E  32
      ISUM=0                                                            E  33
      ITEN=1
      DO 5 J=1,II                                                       E  34
      L=II-J+1                                                          E  35
      ISUM=ISUM+INUM(L)*ITEN                                            E  36
      ITEN=ITEN*10
      INUM(L)=0                                                         E  37
5     CONTINUE                                                          E  38
      IF (ISUM.GT.600) GO TO 9
      NUMBX(MM)=ISUM                                                    E  39
      MM=MM+1                                                           E  40
      II=1                                                              E  41
      GO TO 8                                                           E  42
6     IF ( STRNG    .NE.COMMA) GO TO 8                                  E  43
      IF (PREV.NE.COMMA) GO TO 8                                        E  44
C     TERMINATE CURRENT STRING BUFFER                                   E  45
7     CONTINUE                                                          E  46
      MM=MM-1                                                           E  47
      I=I+1                                                             E  48
      RETURN                                                            E  49
8     PREV= STRNG                                                       E  50
      I=I+1                                                             E  51
      IF (I.LE.80) GO TO 2                                              E  52
      GO TO 1                                                           E  53
```

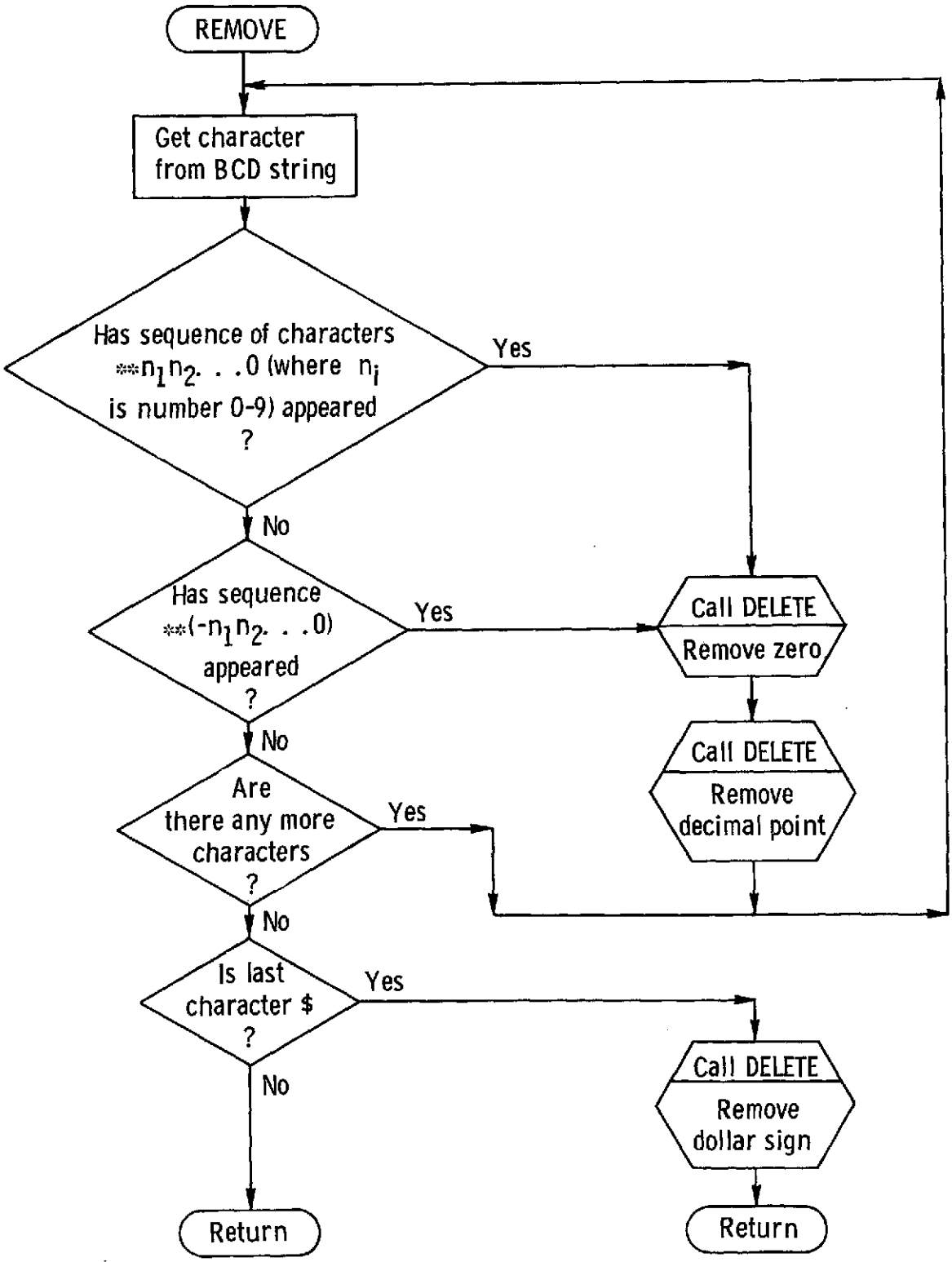
```
C       WRITE ERROR MESSAGES                                                    E  54
9       CONTINUE                                                                E  55
        WRITE (6,11)                                                            E  56
        STOP                                                                    E  57
C                                                                               E  58
10      FORMAT (13A6,A2)                                                        E  60
11      FORMAT (10X,45HSUBSCRIPT TOO LARGE FOR THIS VERSION                )    E  61
        END                                                                     E  62-
```

Subroutine SOUT



```
      SUBROUTINE SOUT (X,ICODE, PRNTP)                                  F   2
      DIMENSION OUT(12)                                                 F   3
      SYMARG X
      DATA BLANK/1H /                                                   F   4
C                                                                       F   5
C     ICODE=1 PRINT                                                     F   6
C     ICODE=2 PUNCH                                                     F   7
C     ICODE=3 PRINT AND PUNCH                                           F   8
C
      C=PRNTP
1     CONTINUE                                                          F   9
      LET Q= BCDCON X,OUT,12                                            F  10
      CALL REMOVE (OUT)                                                 F  11
      IF (ICODE.EQ.1) GO TO 2                                           F  12
      PUNCH 4, (OUT(I),I=2,12)                                          F  13
      IF (ICODE.EQ.2) GO TO 3                                           F  14
2     WRITE(6,5) C,(OUT(I),I=2,12)
      C=BLANK
3     IF (Q.NE.0.) GO TO 1                                              F  16
      RETURN                                                            F  17
C                                                                       F  18
4     FORMAT (5X,1H*,11A6)                                              F  19
5     FORMAT(A1,19X,11A6)
      END                                                               F  21-
```

Subroutine REMOVE



```
      SUBROUTINE REMOVE (X)                                                  1
      INTEGER X(1)                                                           2
      LOGICAL NBR                                                            3
      DATA AAST,AMINUS,AE,ALP,ARP,ADP,AZERO/6H*00000,6H-00000,6HE00000,6     4
     1H(00000,6H)00000,6H.00000,6H000000/,MSW/1/                             5
      DATA ADLR,K/6H$00000,1/                                                6
      LMAX=X(1)+6                                                            7
      L=7                                                                    8
1     GO TO (2,3,4,6),K                                                      9
2     IF (FC(X,L).NE.AAST) GO TO 11                                         10
      GO TO 12                                                              11
3     IF (FC(X,L).NE.AAST) GO TO 11                                         12
      GO TO 13                                                              13
4     IF (FC(X,L).EQ.ALP) GO TO 14                                          14
5     IF (NBR(X,L)) GO TO 7                                                 15
      GO TO 11                                                              16
6     IF (FC(X,L).NE.AMINUS) GO TO 11                                       17
      L=L+1                                                                 18
      MSW=2                                                                 19
      GO TO 5                                                               20
7     L=L+1                                                                 21
      IF (NBR(X,L)) GO TO 7                                                 22
      IF (FC(X,L).NE.ADP) GO TO 11                                          23
      L=L+1                                                                 24
      IF (FC(X,L).NE.AZERO) GO TO 11                                        25
      L=L+1                                                                 26
      GO TO (8,9),MSW                                                       27
8     IF (L.EQ.LMAX+1) GO TO 10                                             28
      IF (NBR(X,L).OR.FC(X,L).EQ.AE) GO TO 11                               29
      GO TO 10                                                              30
9     MSW=1                                                                 31
      IF (L.EQ.LMAX+1) GO TO 10                                             32
      IF (FC(X,L).NE.ARP) GO TO 11                                          33
10    CALL DELETE (X,L-1)                                                   34
      CALL DELETE (X,L-2)                                                   35
11    K=1                                                                   36
      GO TO 15                                                              37
12    K=2                                                                   38
      GO TO 15                                                              39
13    K=3                                                                   40
      GO TO 15                                                              41
14    K=4                                                                   42
15    L=L+1                                                                 43
      IF (L.LE.LMAX) GO TO 1                                                44
      IF (FC(X,LMAX).NE.ADLR) RETURN                                        45
      K=1                                                                   46
      CALL DELETE (X,LMAX)                                                  47
      RETURN                                                                48
      END                                                                   49-
```

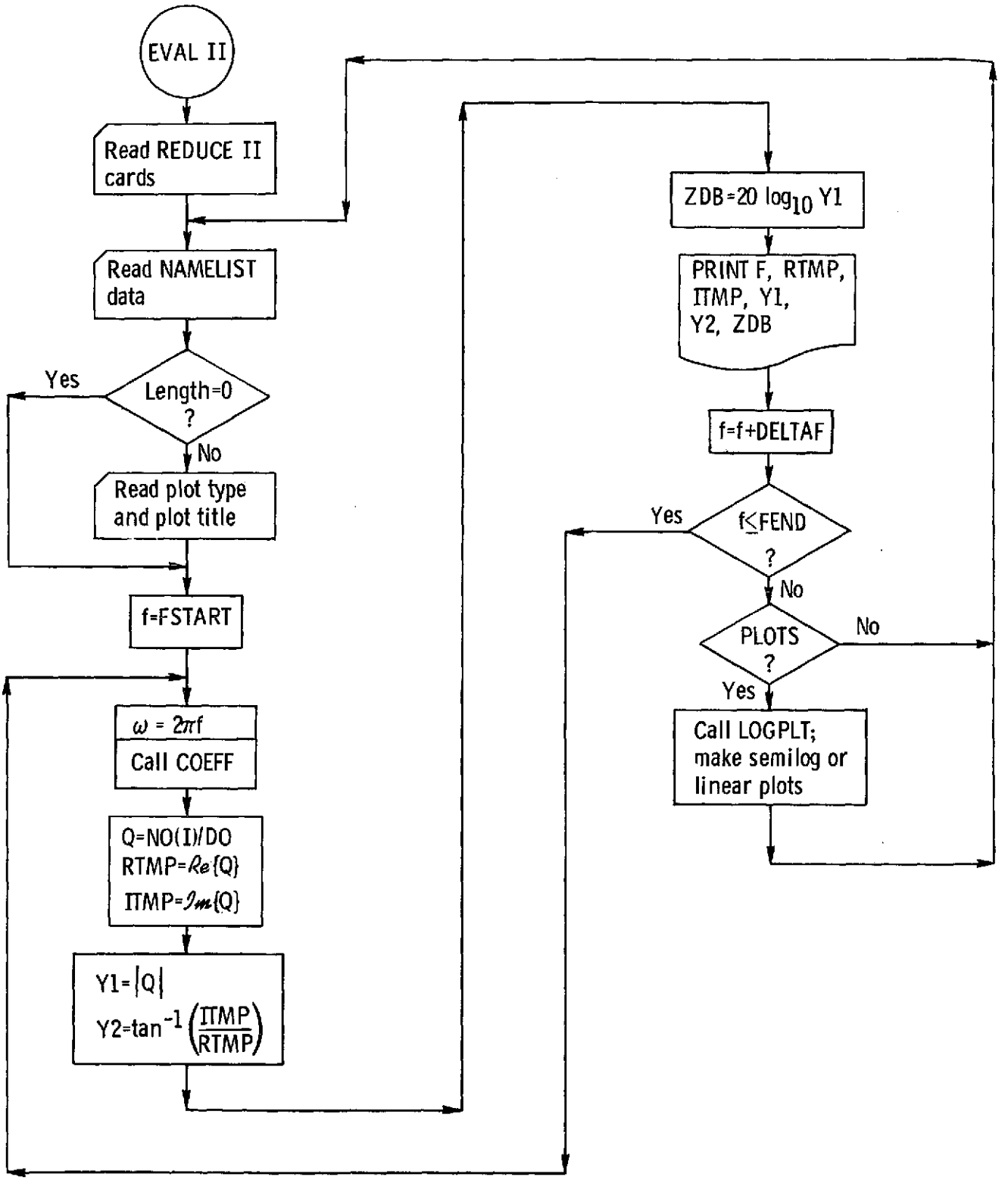
```
      FUNCTION FC (X,L)                                                  1
      DIMENSION X(1)                                                     2
      DATA MASK/0770000000000/                                           3
      LM1=L-1                                                            4
      IW=LM1/6+1                                                         5
      FC=AND(ALS(6*MOD(LM1,6),X(IW)),MASK)                               6
      RETURN                                                             7
      ENTRY FC1(X,L)                                                     8
      FC=AND(ALS(L,X),MASK)                                              9
      RETURN                                                            10
      END                                                               11-
```

```
      SUBROUTINE DELETE (X,L)                                            1
      DIMENSION X(L), ZERO(6), BLANK(6)                                  2
      DATA ZERO/0007777777777,0770077777777,0777700777777,0777777007777,  3
     107777777700777,0777777777700/,BLANK/0600000000000,0006000000000,000  4
     200600000000,0000000600000,0000000006000,0000000000060/            5
      LM1=L-1                                                            6
      IW=LM1/6+1                                                         7
      IP=MOD(LM1,6)+1                                                    8
      X(IW)=OR(AND(X(IW),ZERO(IP)),BLANK(IP))                            9
      RETURN                                                            10
      END                                                               11-
```

```
      LOGICAL FUNCTIONNBR(X,L)                                           1
      REAL X(1)                                                          2
      N=LGR(30,FC(X,L))                                                  3
      NBR=.FALSE.                                                        4
      IF ((N.GE.0).AND.(N.LE.9)) NBR=.TRUE.                              5
      RETURN                                                             6
      END                                                                7-
```

```
      SUBROUTINE COS(X)
      ENTRY SIN(X)
      ENTRY TANH(X)
      ENTRY ATAN(X)
      ENTRY ALOG(X)
      ENTRY DUMP(X)
      ENTRY PDUMP(X)
      ENTRY FMCDIF(X)
      ENTRY EXPDMP(X)
      ENTRY FMCDMP(X)
      STOP
      END
```

Evaluation Program - EVAL II



```
      REAL ITMP,K,LENGTH                                                    1
      INTEGER PLTTYP                                                        2
      COMPLEX NO,DO,Q                                                       3
      COMMON /ARRAY/ BA(14),NNUM,NN                                         4
      COMMON /CEVAL/ T(10),Z(10),WN(10),K(10),NEXP,NSIN,NCOS,NSINH,NCOSH    5
     1,NSQRT                                                                6
      COMMON /OMEGA/ W,DO,NO(5)                                             7
      COMMON /NONLIN/ EXPA(5),SINA(5),COSA(5),EXPB(5),SINB(5),COSB(5),EX    8
     1PC(5),SINC(5),COSC(5),EXPD(5),SIND(5),COSD(5),SINHA(5),COSHA(5),SQ    9
     2RTA(5),SINHB(5),COSHB(5),SQRTB(5),SINHC(5),COSHC(5),SQRTC(5),SINHD    10
     3(5),COSHD(5)                                                          11
      COMMON /CLPLOT/ XPEN,YPEN,NX,NY,IPEN,XLABEL(10),YLABEL(10)            12
      DIMENSION Y1(604,5), Y2(604,5), KKK(6), P(14), X(604)                 13
      DIMENSION NDEN(5)                                                     14
      DATA LENGTH/0./                                                       15
      DATA LOGPLO/6HLOGPLO/                                                 16
      DATA CONVT/57.2957795/                                                17
      DATA YDB/3HDB./                                                       18
      DATA (KKK(J),J=2,6),P(1),(P(L),L=3,14)/50,1,1,0,601,3.,2*0.,10.,2*    19
     10.,10.,5*0.,90./                                                      20
      DATA TWOPI/6.2831853/                                                 21
      DATA XLAB/1HF/                                                        22
      DATA YLAB/3HABS/                                                      23
      DATA ZLAB/5HANGLE/                                                    24
      DATA NK,NWN,NT,NZ/4*0/                                                25
      NAMELIST /INPUT/ NEXP,NSIN,NCOS,EXPA,SINA,COSA,EXPB,SINB,COSB,EXPC    26
     1,SINC,COSC,EXPD,SIND,COSD,NSINH,NCOSH,NSQRT,SINHA,COSHA,SQRTA,SINH    27
     28,COSHB,SQRTB,SINHC,COSHC,SQRTC,SINHD,COSHD,NK,K,NT,T,NZ,Z,NWN,WN,    28
     3FSTART,FEND,DELTAF,LENGTH                                             29
C     READ THE NUMERATOR AND DENOMINATOR INFORMATION SUPPLIED BY REDUCE II
      READ (5,12) NNUM,IK,(NDEN(I),I=1,IK)                                 30
C     READ THE NAMELIST INPUT.  NOTE THAT THE PROGRAM ALWAYS RETURNS HERE
1     READ (5,INPUT)                                                       31
      WRITE (6,18)                                                          32
      IUPPER=1                                                              33
C     CHECK FOR A PLOTTING REQUEST.
      IF (LENGTH.EQ.0.) GO TO 2                                             34
      IUPPER=2                                                              35
C     PLOTTING TO BE DONE.  READ IN PLOT TITLE AND PLOT TYPE
      READ (5,30) BA                                                        36
      WRITE (6,31) BA                                                       37
      READ (5,30) PLTTYP                                                    38
2     CONTINUE                                                              39
C     EXAMINE THE VARIOUS NAMELIST VARIABLE TO GET THE INPUT IDENTIFIED
C     ON THE PROGRAM OUTPUT LISTING
      IF (NEXP.NE.0) WRITE (6,19) (I,EXPA(I),EXPB(I),EXPC(I),EXPD(I),I=1    40
     1,NEXP)                                                                41
      IF (NSIN.NE.0) WRITE (6,20) (I,SINA(I),SINB(I),SINC(I),SIND(I),I=1    42
     1,NSIN)                                                                43
      IF (NCOS.NE.0) WRITE (6,21) (I,COSA(I),COSB(I),COSC(I),COSD(I),I=1    44
     1,NCOS)                                                                45
      IF (NSINH.NE.0) WRITE (6,22) (I,SINHA(I),SINHB(I),SINHC(I),SINHD(I    46
     1),I=1,NSINH)                                                          47
      IF (NCOSH.NE.0) WRITE (6,23) (I,COSHA(I),COSHB(I),COSHC(I),COSHD(I    48
     1),I=1,NCOSH)                                                          49
      IF (NSQRT.NE.0) WRITE (6,24) (I,SQRTA(I),SQRTB(I),SQRTC(I),I=1,NSQ    50
     1RT)                                                                   51
      IF (NK.NE.0) WRITE (6,25) (I,K(I),I=1,NK)                             52
      IF (NT.NE.0) WRITE (6,26) (I,T(I),I=1,NT)                             53
```

```
      IF (NZ.NE.0) WRITE (6,27) (I,Z(I),I=1,NZ)                          54
      IF (NWN.NE.0) WRITE (6,28) (I,WN(I),I=1,NWN)                       55
      WRITE (6,13) (I,NNUM,NDEN(I),I=1,IK)                              56
      WRITE (6,29) FSTART,FEND,DELTAF                                    57
      WRITE (6,14)                                                       58
      IF (DELTAF.EQ.0.0) GO TO 1                                         59
C     START THE MAIN PROGRAM LOOP.  IUPPER =1 IF THERE ARE NO PLOTS
C                                          2 IF PLOTS ARE TO BE MADE
      DO 8 NTYPE=1,IUPPER                                                60
      IF (NTYPE.EQ.2) GO TO 3                                            61
      DF=DELTAF                                                          62
      GO TO 4                                                            63
3     DF=(FEND-FSTART)/600.                                              64
4     J=0                                                                65
      F=FSTART                                                           66
5     W=TWOPI*F                                                          67
      IF (J.GE.604) GO TO 8                                              68
      J=J+1                                                              69
C     THE CALL TO COEFF EVALUATES THE TRANSFER FUNCTION AT S=I*W
      CALL COEFF                                                         70
      X(J)=F                                                             71
      DO 7 I=1,IK                                                        72
      Q=NO(I)/DO                                                         73
      RTMP=REAL(Q)                                                       74
      ITMP=AIMAG(Q)                                                      75
      Y1(J,I)=0.                                                         76
      Y2(J,I)=0.                                                         77
      ZDB=0.                                                             78
      IF (RTMP.EQ.0..AND.ITMP.EQ.0.) GO TO 6                             79
      Y1(J,I)=CABS(Q)                                                    80
      ZDB=20.*ALOG10(Y1(J,I))                                            81
      Y2(J,I)=ATAN2(ITMP,RTMP)*CONVT                                     82
6     CONTINUE                                                           83
      IF (PLTTYP.EQ.LOGPLO.AND.NTYPE.EQ.2) Y1(J,I)=ZDB                   84
      IF (NTYPE.EQ.2) GO TO 7                                            85
      IF (I.EQ.1.AND.IK.GT.1) WRITE (6,15)                               86
      WRITE (6,16) I,F,RTMP,ITMP,Y1(J,I),Y2(J,I),ZDB                     87
7     CONTINUE                                                           88
      F=F+DF                                                             89
      IF (F.LE.FEND) GO TO 5                                             90
8     CONTINUE                                                           91
C     CONTINUE ON WITH THE PLOTTING  IF IT IS CALLED FOR
      IF (LENGTH.EQ.0.0) GO TO 1                                         92
      KKK(6)=J                                                           93
      P(2)=AINT(LENGTH+.5)                                               94
      NX=-1                                                              95
      XLABEL(1)=XLAB                                                     96
      DO 11 I=1,IK                                                       97
      P(6)=0.                                                            98
      P(7)=0.                                                            99
      NY=3                                                              100
      IF (PLTTYP.EQ.LOGPLO) GO TO 9                                     101
      KKK(1)=4                                                          102
      YLABEL(1)=YLAB                                                    103
      GO TO 10                                                          104
9     CONTINUE                                                          105
      KKK(1)=2                                                          106
      IF (I.GT.1) KKK(1)=-2                                             107
      YLABEL(1)=YDB                                                     108
10    CONTINUE                                                          109
```

```
      NN=NDEN(I)                                                        110
C     MAKE THE AMPLITUDE RATIO PLOT
      CALL LOGPLT (X,Y1(1,I),KKK,P)                                     111
      WRITE (6,17)                                                      112
      IF (PLTTYP.EQ.LOGPLO) KKK(1)=-2                                   113
      P(6)=-200.00                                                      114
      P(7)=200.00                                                       115
      YLABEL(1)=ZLAB                                                    116
      NY=5                                                              117
C     MAKE THE PHASE ANGLE PLOT
      CALL LOGPLT (X,Y2(1,I),KKK,P)                                     118
      WRITE (6,17)                                                      119
11    CONTINUE                                                          120
      GO TO 1                                                           121
C                                                                       122
C                                                                       123
12    FORMAT (I5)                                                       124
13    FORMAT (50X,27HNUMBER     TRANSFER FUNCTION,(/,52X,I2,9X,2HX(,I2,4H 125
     1)/X(,I2,1H))%                                                     126
14    FORMAT (//,3X,6HNUMBER,5X,9HFREQUENCY,13X,9HREAL PART,9X,14HIMAGIN 127
     1ARY PART,6X,14HABSOLUTE VALUE,6X,11HPHASE ANGLE,9X,16HAMPLITIDE IN 128
     2 DB.)                                                             129
15    FORMAT (1HK)                                                      130
16    FORMAT (I5,6F20.6)                                                131
17    FORMAT (10X,9HPLOT MADE)                                          132
18    FORMAT (1H1,//,36H FORTRAN EVALUATION OF FORMAC OUTPUT)           133
19    FORMAT (1HO,3X,1HI,5X,5HEXPA ,10X,5HEXPB ,10X,5HEXPC ,10X,5HEXPD , 134
     1/,(I5,4E15.5))                                                    135
20    FORMAT (1HO,3X,1HI,5X,5HSINA ,10X,5HSINB ,10X,5HSINC ,10X,5HSIND , 136
     1/,(I5,4E15.5))                                                    137
21    FORMAT (1HO,3X,1HI,5X,5HCOSA ,10X,5HCOSB ,10X,5HCOSC ,10X,5HCOSD , 138
     1/,(I5,4E15.5))                                                    139
22    FORMAT (1HO,3X,1HI,5X,5HSINHA,10X,5HSINHB,10X,5HSINHC,10X,5HSINHD, 140
     1/,(I5,4E15.5))                                                    141
23    FORMAT (1HO,3X,1HI,5X,5HCOSHA,10X,5HCOSHB,10X,5HCOSHC,10X,5HCOSHD, 142
     1/,(I5,4E15.5))                                                    143
24    FORMAT (1HO,3X,1HI,5X,5HSQRTA,10X,5HSQRTB,10X,5HSQRTC,/,(I5,3E15.5 144
     1))                                                                145
25    FORMAT (1HO,3X,1HI,5X,1HK,/,(I5,E15.5))                           146
26    FORMAT (1HO,3X,1HI,5X,1HT,/,(I5,E15.5))                           147
27    FORMAT (1HO,3X,1HI,5X,1HZ,/,(I5,E15.5))                           148
28    FORMAT (1HO,3X,1HI,5X,2HWN,/,(I5,E15.5))                          149
29    FORMAT (//,8H FSTART=,F15.5,5X,5HFEND=,F15.5,5X,7HDELTAF=,F15.5)  150
30    FORMAT (13A6,A2)                                                  151
31    FORMAT (34HOCALCOMP PLOTS HAVE BEEN REQUESTED,//,5X,13A6,A2)      152
      END                                                               153-
```

```
      SUBROUTINE COEFF                                                      1
C     PUT REDUCE II OUTPUT IN THIS SUBROUTINE                               2
      REAL K                                                                3
C*************************************************************************4
C* *  THE COMPLEX STATEMENT PUNCHED BY REDUCE II FOLLOWS THIS CARD * * *    5
C*************************************************************************6
      LOGICAL DONE                                                          7
      DATA DONE/.FALSE./                                                    8
      COMPLEX NO,DO,EXPFX,SINFX,COSFX,SINHFX,COSHFX,SQRTFX,S                9
      COMPLEX EXPF(5),SINF(5),COSF(5),SINHF(5),COSHF(5),SQRTF(5)           10
      COMMON /CEVAL/ T(10),Z(10),WN(10),K(10),NEXP,NSIN,NCOS,NSINH,NCOSH   11
     1,NSQRT                                                               12
      COMMON /OMEGA/ W,DO,NO(5)                                            13
      IF (NEXP.EQ.0) GO TO 2                                               14
      DO 1 I=1,NEXP                                                        15
1     EXPF(I)=EXPFX(I)                                                     16
2     IF (NSIN.EQ.0) GO TO 4                                               17
      DO 3 I=1,NSIN                                                        18
3     SINF(I)=SINFX(I)                                                     19
4     IF (NCOS.EQ.0) GO TO 6                                               20
      DO 5 I=1,NCOS                                                        21
5     COSF(I)=COSFX(I)                                                     22
6     IF (NSINH.EQ.0) GO TO 8                                              23
      DO 7 I=1,NSINH                                                       24
7     SINHF(I)=SINHFX(I)                                                   25
8     IF (NCOSH.EQ.0) GO TO 10                                             26
      DO 9 I=1,NCOSH                                                       27
9     COSHF(I)=COSHFX(I)                                                   28
10    IF (NSQRT.EQ.0) GO TO 12                                             29
      DO 11 I=1,NSQRT                                                      30
11    SQRTF(I)=SQRTFX(I)                                                   31
12    CONTINUE                                                             32
      S=CMPLX(0.0,W)                                                       33
C*************************************************************************34
C* * * THE USERS'S FUNCTIONS (G(S)'S) ARE INSERTED AFTER THESE CARDS * *  35
C* * *  THE G'S MUST BE FOLLOWED BY THE SUPER G'S,THE NUMERATOR,AND * *   36
C* * *  DENOMINATOR CARDS REDUCE II PUNCHED                   * * *       37
C*************************************************************************38
      RETURN                                                               39
      END                                                                  40-
```

Function EXPFX

```
      COMPLEX FUNCTION EXPFX(I)                                          1
      COMPLEX X,Y                                                        2
      COMMON /OMEGA/ W,DO,NO(5)                                          3
      COMMON /NONLIN/ EXPA(5),SINA(5),COSA(5),EXPB(5),SINB(5),COSB(5),EX 4
     1PC(5),SINC(5),COSC(5),EXPD(5),SIND(5),COSD(5),SINHA(5),COSHA(5),SQ 5
     2RTA(5),SINHB(5),COSHB(5),SQRTB(5),SINHC(5),COSHC(5),SQRTC(5),SINHD 6
     3(5),COSHD(5)                                                       7
      X=CMPLX(EXPC(I)-EXPA(I)*W**2,EXPB(I)*W)                            8
      Y=EXPD(I)*CSQRT(X)                                                 9
      EXPFX=CEXP(Y)                                                     10
      GO TO 1                                                           11
      ENTRY SINFX(I)                                                    12
      X=CMPLX(SINC(I)-SINA(I)*W**2,SINB(I)*W)                           13
      Y=SIND(I)*CSQRT(X)                                                14
      EXPFX=CSIN(Y)                                                     15
      GO TO 1                                                           16
      ENTRY COSFX(I)                                                    17
      X=CMPLX(COSC(I)-COSA(I)*W**2,COSB(I)*W)                           18
      Y=COSD(I)*CSQRT(X)                                                19
      EXPFX=CCOS(Y)                                                     20
      GO TO 1                                                           21
      ENTRY SINHFX(I)                                                   22
      X=CMPLX(SINHC(I)-SINHA(I)*W**2,SINHB(I)*W)                        23
      Y=SINHD(I)*CSQRT(X)                                               24
      X1=REAL(Y)                                                        25
      Y1=AIMAG(Y)                                                       26
      EXPFX=CMPLX(SINH(X1)*COS(Y1),COSH(X1)*SIN(Y1))                    27
      GO TO 1                                                           28
      ENTRY COSHFX(I)                                                   29
      X=CMPLX(COSHC(I)-COSHA(I)*W**2,COSHB(I)*W)                        30
      Y=COSHD(I)*CSQRT(X)                                               31
      X1=REAL(Y)                                                        32
      Y1=AIMAG(Y)                                                       33
      EXPFX=CMPLX(COSH(X1)*COS(Y1),SINH(X1)*SIN(Y1))                    34
      GO TO 1                                                           35
      ENTRY SQRTFX(I)                                                   36
      X=CMPLX(SQRTC(I)-SQRTA(I)*W**2,SQRTB(I)*W)                        37
      EXPFX=CSQRT(X)                                                    38
      RETURN                                                            39
      END                                                               40-
```

Subroutine CALTIT

```
SUBROUTINE CALTIT                                                    1
COMMON /ARRAY/ BA(14),NDEN,NNUM                                      2
DIMENSION FUNC(4)                                                    3
DATA FUNC(1)/20HTRANSFER FUNCTION X /                                4
AD=NDEN                                                              5
AN=NNUM                                                              6
CALL SYMBOL (-1.5,1.,.15,BA,90.,80)                                  7
CALL SYMBOL (-1.0,1.,.15,FUNC,90.,19)                                8
CALL NUMBER (-.85,3.4,.100,AD,90.,-1)                                9
CALL SYMBOL (-1.,3.52,.15,2H/X,90.,2)                               10
CALL NUMBER (-.85,3.70,.100,AN,90.,-1)                              11
RETURN                                                              12
END                                                                 13-
```

Block Data

```
BLOCK DATA
COMMON /CLPLOT/ XPEN,YPEN,NX,NY,IPEN,XL(10),YL(10)
DATA XPEN,YPEN,IPEN/2*0.,-3/
END
```

```
      SUBROUTINE LOGPLT (XDOWN,YACROS,KKK,P)                                1
C                                                                           2
C     THIS LOG PLOTTING PROGRAM WAS WRITTEN BY MR. BERT HENRY OF            3
C     LEWIS RESEARCH CENTER.  IT WAS MODIFIED BY MR. JERRY LENHART AND      4
C     MR. JOHN RIEHL ALSO OF LEWIS RESEARCH CENTER                          5
C                                                                           6
C   THIS IS A GENERAL ROUTINE  FOR MAKING LOG-LOG AND SEMI-LOG CALCOMP      7
C             PLOTS ON PLAIN PAPER.                                         8
C                                                                           9
C   THIS SUBROUTINE UTILIZES THE SAME CONVENTIONS AS CALPLT EXCEPT FOR      10
C             KKK(1),P(3),P(4),P(6), AND P(7) AS INDICATED BELOW.           11
C             THE ROUTINE EXAMINES KKK(1) TO DETERMINE WHICH AXIS WILL      12
C             REQUIRE A LOG SCALE, THEN EXAMINES THE APPROPRIATE DATA       13
C             TO DETERMINE THE NUMBER OF CYCLES REQUIRED.  IF THE NUMBER    14
C             OF CYCLES DOES NOT EXCEED 7 OR 15 ( Y OR X AXIS RE-           15
C             SPECTIVELY ) , THE GRIDS ARE DRAWN                            16
C             PERPENDICULAR TO THE LOG AXIS AT EVEN VALUES IN EACH CYCLE    17
C             IF MORE THAN 7 OR 15 CYCLES ARE REQUIRED, AN APPROPRIATE      18
C             MESSAGE IS PRINTED AND CONTROL IS RETURNED TO THE CALL-       19
C             ING PROGRAM.                                                  20
C                                                                           21
C   FOR THE LINEAR SCALE, THE ROUTINE WILL COMPUTE THE SCALE IF AND ONLY     22
C             IF BOTH P(3) AND P(4), OR P(6) AND P(7), ARE SET TO 0.0       23
C             FOR THE X OR Y AXIS RESPECTIVELY.  GRID LINES WILL BE         24
C             DRAWN PERPENDICULAR TO THE AXIS AT THE TICK MARKS WHICH       25
C             ARE CONTROLLED BY P(8).                                       26
C                                                                           27
C   A DESCRIPTION OF THE SUBROUTINES ARGUMENTS FOLLOWS-,                    28
C   XDOWN IS  THE NAME OF THE ARRAY CONTAINING THE VARIABLE TO BE PLOTTED   29
C   AS THE ABSCISSA                                                         30
C   YACROS = THE NAME OF THE ARRAY CONTAINING THE VARIABLE TO BE PLOTTED    31
C   THE ORDINATE                                                            32
C                                                                           33
C   KKK(1) = 1,  IF   Y  AXIS  ONLY    IS LOGARITHMIC                       34
C   KKK(1) = 2,  IF   X  AXIS  ONLY    IS LOGARITHMIC                       35
C   KKK(1) = 3,  IF BOTH X AND Y AXES ARE LOGARITHMIC                       36
C   KKK(1) = 4,  IF BOTH X AND Y AXES ARE LINEAR                            37
C    **** IF KKK(1) IS NEGATIVE THE APPROPRIATE NON-LINEAR DATA IS          38
C         ASSUMED TO BE IN LOG BASE 10 FORM. IF POSITIVE THEN THE DATA      39
C         WILL BE TRANSFORMED TO LOG BASE 10 AND LEFT IN THIS FORM.         40
C KKK(2) = USED TO CONTROL SYMBOLS PLACED ON DATA POINTS                    41
C KKK(2) 0, FOR ONLY A LINE PLOT                                            42
C KKK(2) 1,2,3, ETC, WILL PRODUCE A SYMBOL AT EVERY DATA POINT, OR          43
C         EVERY KKK(2) DATA POINTS                                          44
C----   A NEGATIVE KKK(2) WILL SUPPRESS THE LINE BETWEEN DATA POINTS        45
C   KKK(3) = THE NUMBER OF CURVES TO BE PLOTTED (KN)                        46
C   KKK(4) = THE REPEAT CYCLE IN CASE VALUES ARE IN A MIXED ARRAY           47
C   KKK(5) = THE DESIRED SYMBOL                                             48
C   KKK(6) - KKK(KN+5) = THE NUMBER OF POINTS IN EACH CURVE                 49
C   P(1) = 1.0, OPTION FOR DUPX, AS USED IN PLOTMY                          50
C   P(1) = 2.0, OPTION FOR DUPY, AS USED IN PLOTMY                          51
C   P(1) = 3.0, OPTION FOR NODUP, AS USED IN PLOTMY                         52
C   P(2) = THE MAXIMUM WIDTH OF THE PLOTTED LINE                            53
C   *****NOTE*****                                                         54
C   IF ABSCISSA VALUES ARE LINEAR, THEN                                     55
C   P(3) = MINIMUM VALUE OF ABSCISSA ARRAY OF DATA                          56
C   P(4) = MAXIMUM VALUE OF ABSCISSA ARRAY OF DATA                          57
C     IF ABS(KKK(1)) = 1 AND EITHER P(3) OR P(4) IS NON-ZERO, THEN THE X    58
```

```
C               SCALE WILL NOT  BE COMPUTED BY THE PLOT ROUTINE.
C   P(5) = THE MAXIMUM HEIGHT OF THE PLOTTED
C               MUST NOT EXCEED 10. INCHES.
C   *****NOTE*****
C   IF ORDINATE VALUES ARE LINEAR, THEN
C   P(6) = THE MINIMUM VALUE OF THE ORDINATE ARRAY OF DATA
C   P(7) = THE MAXIMUM VALUE OF THE ORDINATE ARRAY OF DATA
C     IF ABS(KKK(1)) = 2 AND EITHER P(6) OR P(7) IS NON-ZERO, THEN THE Y
C               SCALE WILL NOT  BE COMPUTED BY THE PLOT ROUTINE.
C   *****NOTE*****
C   IF EITHER X OR Y ARE LINEAR, THEN
C   P(8) = DIV, (WILL CAUSE A TICK MARK TO BE PLACED AT EVERY 10.0/DIV IN
C   **************
C   P(9) - P(10),= THE COORDINATES OF THE STARTING POINT OF THE ABSCISSA
C   P(11) - P(12),= THE COORDINATES OF THE STARTING POINT OF THE ORDINATE
C   P(13) = THE ANGLE IN (DEGREES) OF THE ABCISSA AXIS
C   P(14) = THE ANGLE IN (DEGREES) OF THE ORDINATE AXIS
C   P(15) = NON-ZERO IF DATA IS TO BE ROTATED PER P(13) AND P(14)
C
C   NOTE********
C   THE FOLLOWING CARD IS NEEDED IN THE CALLING PROGRAM
C      COMMON /CLPLOT/XPEN,YPEN,NX,NY,IPEN,XLABEL(10),YLABEL(10)
C   WHERE-
C       XPEN AND YPEN GENERATES THE INCREMENTS NECESSARY TO MOVE FROM CUR
C       POSITION TO XPEN,YPEN-----------------NORMALLY=0.0
C   IPEN=3, FOR PEN UP
C   IPEN=2, FOR PEN DOWN
C  IPEN=-3 OR-2 ,PROVIDES A NEW REFERENCE POINT AT (XPEN,YPEN) RELATIVE
C       THE CURRENT REFERENCE POINT
C   TITLES
C   XLABEL = THE NAME OF THE ALPHABETIC ARRAY OF DATA FOR THE ORDINATE AX
C   YLABEL = THE NAME OF THE ALPHABETIC ARRAY OF DATA FOR THE ABCISSA AXI
C   NX= THE NUMBER OF CHARACTERS IN THE ORDINATE AXIS TITLE, A MINUS NX P
C       THE ANNOTATION ON THE CLOCKWISE SIDE OF THE AXIS INSTEAD OF THE
C       COUNTER CLOCKWISE SIDE
C   NY= THE NUMBER OF CHARACTERS IN THE ABCISSA  AXIS TITLE, A MINUS NY P
C       THE ANNOTATION ON THE CLOCKWISE SIDE OF THE AXIS INSTEAD OF THE
C       COUNTER CLOCKWISE SIDE
      COMMON /CLPLOT/ XPEN,YPEN,NX,NY,IPEN,XLABEL(10),YLABEL(10)
      COMMON /SPECL/ TEST,ORGSET,SPASET
      COMMON /XCPIDX/ NBLCNO,IRUNO,CPID(8)
      DIMENSION SAVARA(4)
      DIMENSION XDOWN(1), YACROS(1), KKK(1), P(1)
      EQUIVALENCE (STR,ISTR)
      DATA RUNMES/6HRUN NC/
      DATA BLKMES/6HBLK NC/
      DATA INITAL/50/
      DATA STR/05C0000000000/
      DATA SPACER/8./
      IF (INITAL.EQ.50) GO TO 3
1     IF (P(1).EQ.4.0) GO TO 2
      IF (IPEN.NE.(-3)) GO TO 5
      IF (XPEN.EQ.STR) XPEN=0.
      IF (YPEN.EQ.STR) YPEN=0.
      GO TO 4
2     CALL PLOT (XPEN,YPEN,IPEN)
      RETURN
3     XTPEN=0.0
      YTPEN=0.0
      ITPEN=-3
```

```
      XSPAC=SPACER                                                      119
      IF (P(1).EQ.STR) P(1)=3.0                                         120
      IF (P(2).EQ.STR) P(2)=10.0                                        121
      IF (P(3).EQ.STR) P(3)=0.                                          122
      IF (P(4).EQ.STR) P(4)=0.                                          123
      IF (P(5).EQ.STR) P(5)=10.0                                        124
      IF (P(6).EQ.STR) P(6)=0.                                          125
      IF (P(7).EQ.STR) P(7)=0.                                          126
      IF (P(8).EQ.STR) P(8)=10.0                                        127
      IF (P(9).EQ.STR) P(9)=0.                                          128
      IF (P(10).EQ.STR) P(10)=0.                                        129
      IF (P(11).EQ.STR) P(11)=0.                                        130
      IF (P(12).EQ.STR) P(12)=0.                                        131
      IF (P(13).EQ.STR) P(13)=0.                                        132
      IF (P(14).EQ.STR) P(14)=90.0                                      133
      IF (P(15).EQ.STR) P(15)=0.                                        134
      IF (KKK(1).EQ.ISTR) KKK(1)=1                                      135
      IF (KKK(2).EQ.ISTR) KKK(2)=0                                      136
      IF (KKK(3).EQ.ISTR) KKK(3)=1                                      137
      IF (KKK(4).EQ.ISTR) KKK(4)=1                                      138
      IF (KKK(5).EQ.ISTR) KKK(5)=1                                      139
      IF (KKK(6).EQ.ISTR) KKK(6)=1                                      140
      INITAL=0                                                          141
      CALL PLOT (XTPEN,YTPEN,ITPEN)                                     142
      FBLONO=NBLONO                                                     143
      CALL SYMBOL (0.0,1.0,0.2,CPID(1),90.0,24)                         144
      CALL SYMBOL (0.5,1.0,0.2,RUNMES,90.0,6)                           145
      CALL SYMBOL (0.5,2.5,0.2,IRUNO,90.0,6)                            146
      CALL SYMBOL (0.5,4.0,0.20,BLKMES,90.0,6)                          147
      CALL NUMBER (0.5,6.0,0.20,FBLONO,90.0,-1)                         148
      XTPEN=10.0                                                        149
      CALL PLOT (XTPEN,YTPEN,ITPEN)                                     150
      GO TO 1                                                           151
4     CALL PLOT (XPEN,YPEN,IPEN)                                        152
5     NX1=-1                                                            153
      FBLONO=NBLONO                                                     154
      CALL SYMBOL (-3.5,0.0,0.15,BLKMES,0.0,6)                          155
      CALL NUMBER (-2.5,0.0,0.15,FBLONO,0.0,-1)                         156
      YTPEN=0.0                                                         157
      XTPEN=2.0                                                         158
      ITPEN=-3                                                          159
      CALL PLOT (XTPEN,YTPEN,ITPEN)                                     160
C CHECK FOR SCALING IN DUPX OPTION                                      161
      JX=1                                                              162
      JY=1                                                              163
      IF (P(3).NE.0.) GO TO 6                                           164
      IF (P(4).EQ.0.) JX=2                                              165
6     IF (P(6).NE.0.) GO TO 7                                           166
      IF (P(7).EQ.0.) JY=2                                              167
7     J=IABS(KKK(1))                                                    168
      IPOPT=P(1)                                                        169
      KN=KKK(3)                                                         170
      NPTS=0                                                            171
      DO 8 I=1,KN                                                       172
8     NPTS=NPTS+KKK(I+5)                                                173
C  TRANSLATE AND ROTATE DATA IF DESIRED                                174
      IF (P(3).NE.3.) GO TO 9                                           175
C     IF (P(15).EQ.0.) GO TO 63                                         176
      CALL TRNSFM (XDOWN,YACROS,NPTS,P(13),P(14))                       177
C     SETUP NUMBER OF POINTS FOR DUPX, AND DUPY                         178
```

```
5       NSPTS=KN*KKK(6)                                                 179
        NPT=KKK(6)                                                      180
        IF (KKK(6).EQ.0) GO TO 29                                       181
        GO TO (10,11,12),IPOPT                                          182
C SETUP FOR AXIS SUBROUTINE                                             183
C       CHECK FOR DUPX,DUPY OR NODUP OPTION                             184
10      NP1=NPT                                                         185
        NP2=NSPTS                                                       186
        GO TO 13                                                        187
11      NP1=NSPTS                                                       188
        NP2=NPT                                                         189
        GO TO 13                                                        190
12      NP1=NPTS                                                        191
        NP2=NPTS                                                        192
13      GO TO (14,19,18,14),J                                           193
14      GO TO (15,16),JX                                                194
15      XMIN=P(3)                                                       195
        DELX=(P(4)-P(3))/P(2)                                           196
        GO TO 17                                                        197
16      CALL SCALE (XDOWN,P(2),NP1,KKK(4),P(8),XMIN,DELX)               198
17      CALL AXGRID (P(9),P(10),XLABEL,NX,P(2),P(13),XMIN,DELX,P(8),P(5),P  199
     1(14))                                                             200
        IF (J.EQ.4) GO TO 19                                            201
18      CALL GRID (P(11),P(12),YLABEL,NY,YACROS,NP2,KKK(1),P(5),P(14),P(2)  202
     1,P(13),YMIN,DELY,7)                                               203
        GO TO (24,24,23,24),J                                           204
19      GO TO (20,21),JY                                                205
20      YMIN=P(6)                                                       206
        DELY=(P(7)-P(6))/P(5)                                           207
        GO TO 22                                                        208
21      CALL SCALE (YACROS,P(5),NP2,KKK(4),P(8),YMIN,DELY)              209
22      CALL AXGRID (P(11),P(12),YLABEL,NY,P(5),P(14),YMIN,DELY,P(8),P(2),  210
     1P(13),15)                                                         211
        IF (J.EQ.4) GO TO 24                                            212
23      CALL GRID (P(9),P(10),XLABEL,NX,XDOWN,NP1,KKK(1),P(2),P(13),P(5),P  213
     1(14),XMIN,DELX)                                                   214
C SETUP FOR ENTERING LINE AND AXIS SUBROUTINES                          215
24      CONTINUE                                                        216
        K2=0                                                            217
        JJJ2=0                                                          218
        DO 28 IND=1,KN                                                  219
        K3=IND                                                          220
        K4=K2                                                           221
        IKSYMB=KKK(5)+IND-1                                             222
        ISYM=2                                                          223
        IF (KKK(2).EQ.999) ISYM=KN+K3+5                                 224
C SETUP FOR CHECKING DUPX,DUPY,OR NODUP OPTION                          225
C                                                                       226
        GO TO (25,26,27),IPOPT                                          227
C DUPX OPTION FOR LINE                                                  228
C                                                                       229
25      IF (KKK(6).EQ.0) GO TO 28                                       230
        K2=K2+KKK(6)                                                    231
        CALL LINE (XDOWN,YACROS(K4+1),KKK(6),KKK(4),KKK(ISYM),IKSYMB,XMIN,  232
     1DELX,YMIN,DELY)                                                   233
        GO TO 28                                                        234
C DUPY OPTION FOR LINE                                                  235
C                                                                       236
26      IF (KKK(6).EQ.0) GO TO 28                                       237
        K2=K2+KKK(6)                                                    238
```

```
      CALL LINE (XDOWN(K4+1),YACROS,KKK(6),KKK(4),KKK(ISYM),IKSYMB,XMIN,     239
     1DELX,YMIN,DELY)                                                        240
      GO TO 28                                                               241
C NODLP OPTION FOR LINE                                                      242
C                                                                            243
27    IF (KKK(K3+5).EQ.0) GO TO 28                                           244
      K2=K2+KKK(K3+5)                                                        245
      K1=K2                                                                  246
      JJJ2=K2+2                                                              247
      CALL LINE (XDOWN(K4+1),YACROS(K4+1),KKK(K3+5),KKK(4),KKK(ISYM),IKS     248
     1YMB,XMIN,DELX,YMIN,DELY)                                               249
28    CONTINUE                                                               250
      CALL CALTIT                                                            251
      XPEN1=P(2)+XSPAC                                                       252
      IPEN=XPEN1                                                             253
      XPEN1=IPEN                                                             254
      YPEN=0.0                                                               255
      IPEN=-3                                                                256
      CALL PLOT (XPEN1,YPEN,IPEN)                                            257
      XSPAC=SPACER                                                           258
29    RETURN
      END                                                                    259-
```

```
C    THIS SUBROUTINE WILL DRAW AXIS  WITH LOG SCALE AND ITS GRID LINES
     SUBROUTINE GRID (X,Y,BCD,NN,XX,N,M,SIZE,THETA,HGT,THTY,XXMIN,XXDEL
    1,KCYMAX)
C    WHERE-  X,Y    IS THE COORDINATE OF THE STARTING POINT OF THE AXIS.
C            BCD    IS THE LOCATION OF ALPHA INFORMATION FOR AXIS TITLE.
C            NN     IS THE NUMBER OF CHARACTERS IN BCD TITLE. A NEGATIVE
C                   NN PLACES THE TITLE ON THE CLOCKWISE SIDE OF THE AXIS
C                   LINE AND VICE-VERSA
C            XX     IS THE NAME OF THE ARRAY IN THE AXIS DIRECTION
C            N      IS THE NUMBER OF POINTS GIVEN IN THE XX ARRAY
C            M      IS POSITIVE IF THE XX ARRAY IS TO BE CONVERTED TO LOG
C                   BASE 10. FOR NEGATIVE M THE XX ARRAY VALUES WILL BE
C                   TAKEN AS LOG BASE 10 VALUES
C            SIZE   IS THE LENGTH OF THE AXIS TO BE PLOTTED IN INCHES
C            THETA  IS THE ANGLE IN DEGREES OF THE AXIS MEASURED IN
C                   COUNTER-CLOCKWISE DIRECTION FROM THE X AXIS.
C            HGT    IS THE LENGTH OF THE OTHER AXIS IN INCHES
C            THTY   IS THE ANGLE IN DEGREES OF THE OTHER AXIS MEASURED
C                   COUNTER-CLOCKWISE FROM THE X AXIS.
C
      DIMENSION XX(1), TIC(9), XSPC(9), YSPC(9), VAL(5)
      DATA (TIC(I),I=1,9)/.0457575,.0511525,.057992,.0669467,.0791813,.0
     19691,.1249387,.1760913,.30103/
      DATA (VAL(I),I=1,5)/8.,6.,4.,2.,1./
      DATA RAD/.0174532925/
      KATE=1
      XMIN=1.E28
      XMAX=-XMIN
      DO 2 I=1,N
      IF (M.LT.0) GO TO 1
      XX(I)=ALOG10(XX(I))
1     IF (XX(I).GT.XMAX) XMAX=XX(I)
      IF (XX(I).LT.XMIN) XMIN=XX(I)
2     CONTINUE
      KXMAX=XMAX
      XMAXK=KXMAX
      IF (XMAX.GT.XMAXK) XMAXK=XMAXK+1.
      KXMIN=XMIN
      XMINK=KXMIN
      IF (XMIN.LT.XMINK) XMINK=XMINK-1.
      KXCYC=XMAXK-XMINK
      IF (KXCYC.GT.7.AND.KCYMAX.EQ.7) GO TO 13
      IF (KXCYC.GT.15.AND.KCYMAX.EQ.15) GO TO 14
      SCALEX=SIZE/FLOAT(KXCYC)
      XXMIN=XMINK
      XXDEL=VAL(5)/SCALEX
      NCX=5*KXCYC
      NCXPTS=9*KXCYC
      CTH=THETA*RAD
      STH=SIN(CTH)
      CTH=COS(CTH)
      CTHY=(THTY-90.)*RAD
      STHY=SIN(CTHY)
      CTHY=COS(CTHY)
      DO 3 I=1,9
      XSPC(I)=TIC(I)*CTH*SCALEX
3     YSPC(I)=TIC(I)*STH*SCALEX
      XN=X+CTH*SIZE
```

```
      YN=Y+STH*SIZE                                                    59
      XNX=XN                                                           60
      YNY=YN                                                           61
      CALL PLOT (XN,YN,3)                                              62
      DO 4 I=1,NOXPTS                                                  63
      JJ=I-1                                                           64
      KK=MOD(JJ,9)+1                                                   65
      XN=XN-XSPC(KK)                                                   66
      YN=YN-YSPC(KK)                                                   67
4     CALL SYMBOL (XN,YN,.1,13,THETA,-2)                               68
      DO 5 I=1,4                                                       69
      XSPC(I)=XSPC(2*I-1)+XSPC(2*I)                                    70
5     YSPC(I)=YSPC(2*I-1)+YSPC(2*I)                                    71
      XSPC(5)=XSPC(9)                                                  72
      YSPC(5)=YSPC(9)                                                  73
      XNU=XNX-STHY*HGT                                                 74
      YNU=YNY+CTHY*HGT                                                 75
      KOTO=1                                                           76
      IF (ABS(XSPC).LT.1.E-3) KOTO=2                                   77
      DO 8 I=1,NOX                                                     78
      JJ=I-1                                                           79
      KK=MOD(JJ,5)+1                                                   80
      CALL PLOT (XNU,YNU,3)                                            81
      CALL PLOT (XNX,YNY,2)                                            82
      XNX=XNX-XSPC(KK)                                                 83
      YNY=YNY-YSPC(KK)                                                 84
      XNU=XNU-XSPC(KK)                                                 85
      YNU=YNU-YSPC(KK)                                                 86
      GO TO (6,7),KOTO                                                 87
6     TEMM=YNU                                                         88
      YNU=YNY                                                          89
      YNY=TEMM                                                         90
      GO TO 8                                                          91
7     TEMM=XNU                                                         92
      XNU=XNX                                                          93
      XNX=TEMM                                                         94
8     CONTINUE                                                         95
      CALL PLOT (0.,0.,3)                                              96
      NCTE=NOX+1                                                       97
      DXB=-.05                                                         98
      DYB=-.05*(1.-3.*STH)+SIGN(.15,NN)                                99
      DXC=SIZE/2.-(.12*FLOAT((IABS(NN)+7)/2))                         100
      DYC=(-.075+SIGN(.375,NN))                                       101
      XN=X+DXB*CTH-DYB*STH                                            102
      YN=Y+DYB*CTH+DXB*STH                                            103
      SPC=.20                                                         104
      SPS=.06                                                         105
      XNT=SQRT(DXC**2+DYC**2)                                         106
      DO 12 I=1,NCTE                                                  107
      JJ=I-1                                                          108
      KK=MOD(JJ,5)                                                    109
      LL=5-KK                                                         110
      IF (KK.NE.0) GO TO 9                                            111
      CALL SYMBOL (XN,YN,.1,2H10,0.,2)                                112
      XT=XN+SPC                                                       113
      YT=YN+SPS                                                       114
      CALL NUMBER (XT,YT,.07,XMINK,0.,-1)                             115
      XMINK=XMINK+1.                                                  116
      GO TO 10                                                        117
9     CALL NUMBER (XN,YN,.1,VAL(LL),0.,-1)                            118
```

```
10      XN=XN+XSPC(LL)                                                    119
        YN=YN+YSPC(LL)                                                    120
        GO TO (11,12),KATE                                                121
11      IF (XNT.GT.SQRT(XN**2+YN**2)) GO TO 12                            122
        XT=X+DXC*CTH-DYC*STH                                              123
        YT=Y+DYC*CTH+DXC*STH                                              124
        CALL SYMBOL (XT,YT,.14,BCD,THETA,IABS(NN))                        125
        KATE=2                                                            126
12      CONTINUE                                                          127
        RETURN                                                            128
13      WRITE (6,17)                                                      129
        GO TO 15                                                          130
14      WRITE (6,16)                                                      131
15      CONTINUE                                                          132
        STOP                                                              133
C                                                                         134
16      FORMAT (33HCNLY 15 CYCLES ALLOWED FOR X-AXIS)                     135
17      FORMAT (32HCNLY 7 CYCLES ALLOWED FOR Y-AXIS)                      136
        END                                                               137-
```

Subroutine TRNSFM

```
      SUBROUTINE TRNSFM (X,Y,N,THETX,THETY)                               1
      DIMENSION X(1), Y(1)                                                2
      RETURN                                                              3
      END                                                                 4-
```

Subroutine AXGRID

```
C    THIS SUBROUTINE WILL DRAW A LINEAR AXIS WITH ITS GRID LINES                1
       SUBROUTINE AXGRID (X,Y,BCD,N,SIZE,THETA,XMIN,DX,DV,HGT,THTY)             2
C                                                                               3
C    WHERE-  X,Y    IS THE COORDINATE OF THE STARTING POINT OF THE AXIS.        4
C                   BOTH ARE FLOATING POINT AND PAGE INCHES.                    5
C            BCD    IS THE LOCATION OF ALPHA INFORMATION FOR AXIS TITLE.        6
C                   USUALLY VARIABLE NAME. (NORMALLY SET UP WITH LITERAL)       7
C            N      IS THE NUMBER OF CHARACTERS IN BCD TITLE. A NEGATIVE        8
C                   N PLACES THE ANNOTATION ON THE CLOCKWISE SIDE OF AXIS       9
C                   LINE AND VICE-VERSA.                                       10
C            SIZE   IS THE LENGTH OF THE AXIS TO BE DRAWN. SIZE IS             11
C                   FLOATING POINT AND SHOULD BE MULTIPLY OF (10.0/DIV).       12
C            THETA  IS THE ANGLE OF THE AXIS MEASURED COUNTER-CLOCKWISE        13
C                   FROM X AXIS. THETA IS FLOATING POINT DEGREES.              14
C            XMIN   IS THE VALUE OF VARIABLE AT THE FIRST POINT OF THE         15
C                   AXIS. XMIN IS FLOATING POINT.  (SEE NOTE)                  16
C            DX     IS THE DIFFERENCE BETWEEN SECOND AND FIRST VALUE OF        17
C                   VARIABLE ALONG AXIS. DX IS FLOATING POINT. (SEE NOTE)      18
C            DV     IS THE NUMBER OF DIVISION PER INCH OF PAPER TO BE USED     19
C                   DV  IS FLOATING POINT. (MAY BE 10.0,20.0,25.0,25.4)        20
C            HGT    IS THE LENGTH OF THE OTHER AXIS                            21
C            THTY   IS THE ANGLE OF THE OTHER AXIS TO BE USED.                 22
C    NOTE- THE SECOND VERSION OF SCALE PLACES XMIN IN VARIABLE(J*K+1)AND        23
C          DX IN VARIABLE(J*K+K+1). WHERE J IS NUMBER OF ELEMENTS IN           24
C          ARRAY AND K IS THE REPEAT CYCLE OF MIXED ARRAY.                     25
C                                                                              26
      DIV=DV                                                                   27
      TWO=2.0                                                                  28
      IF (DIV.LT.1..OR.DIV.GT.25.4) DIV=10.0                                   29
      TH=THETA*0.0174532925                                                    30
      THETY=(THTY-90.)*.0174532925                                             31
      STHY=SIN(THETY)                                                          32
      CTHY=COS(THETY)                                                          33
      CTH=COS(TH)                                                              34
      STH=SIN(TH)                                                              35
      DIVA=DIV                                                                 36
      IF (DIV-20.0) 1,2,2                                                      37
1     DIVA=2.0*DIV                                                             38
      TWO=1.0                                                                  39
2     SPACE=10.0/DIVA                                                          40
      DXB=-.05*(1.+2.*CTH)                                                     41
      DYB=-.05*(1.-5.*STH)+SIGN(.15,N)                                         42
      DXC=SIZE/2.-(.12*FLOAT((IABS(N))/2))                                     43
      DYC=(-.075+SIGN(.575,N))                                                 44
      DNOTE=SIZE/SPACE/2.0+1.0                                                 45
      NOTE=DNOTE                                                               46
      IF ((DNOTE-FLOAT(NOTE)).GT.0.99) NOTE=NOTE+1                             47
      SPC=CTH*SPACE*2.0                                                        48
      SPS=STH*SPACE*2.0                                                        49
      XN=X+DXB*CTH-DYB*STH                                                     50
      YN=Y+DYB*CTH+DXB*STH                                                     51
      ADY=ABS(DX)*10.0/DIV                                                     52
      EX=0.0                                                                   53
      IF (ADY) 3,7,3                                                           54
3     IF (ADY-100.0) 6,4,4                                                     55
4     ADY=ADY/10.0                                                             56
      EX=EX+1.0                                                                57
      GO TO 3                                                                  58
```

```
5       ADY=ADY*10.0                                                          59
        EX=EX-1.0                                                             60
6       IF (ADY-1.) 5,7,7                                                     61
7       ABSV=XMIN/10.0**EX                                                    62
        NADY=ADY                                                              63
        IF (ABS(ADY-FLOAT(NADY)).GT..99) ADY=IFIX(ADY+SIGN(.01,ADY))          64
        ADY=SIGN(ADY,DX)*TWC                                                  65
        NT=SQRT(DXC**2+DYC**2)                                                66
        DO 10 I=1,NCTE                                                        67
        CALL NUMBER (XN,YN,.1,ABSV,0.,0)                                      68
        ABSV=ABSV+ADY                                                         69
        XN=XN+SPC                                                             70
        YN=YN+SPS                                                             71
        IF (NT) 10,8,10                                                       72
8       XT=X+DXC*CTH-DYC*STH                                                  73
        YT=Y+DYC*CTH+DXC*STH                                                  74
        CALL SYMBOL (XT,YT,.14,BCD,THETA,IABS(N))                             75
        IF (EX) 9,10,9                                                        76
9       XT=XT+.12*FLOAT(IABS(N))*CTH                                          77
        YT=YT+.12*FLOAT(IABS(N))*STH                                          78
        CALL SYMBOL (XT,YT,.14,7H(X10  ),THETA,7)                             79
        XT=XT+(.12*5.*CTH-.06*STH)                                            80
        YT=YT+(.12*5.*STH+.06*CTH)                                            81
        CALL NUMBER (XT,YT,.08,EX,THETA,-1)                                   82
10      NT=NT-1                                                               83
        DNTIC=DIV*SIZE/10.0+1.0                                               84
        NTIC=DNTIC                                                            85
        IF ((DNTIC-FLOAT(NTIC)).GT.0.99) NTIC=NTIC+1                          86
        XN=X+CTH*SIZE                                                         87
        YN=Y+STH*SIZE                                                         88
        XNX=XN                                                                89
        YNY=YN                                                                90
        XNU=XNX-STHY*HGT                                                      91
        YNU=YNY+CTHY*HGT                                                      92
        SPC=CTH*10.0/DIV                                                      93
        SPS=STH*10.0/DIV                                                      94
        CALL PLOT (XN,YN,3)                                                   95
        XN=FLOAT(NTIC-1)*SPC+X                                                96
        YN=FLOAT(NTIC-1)*SPS+Y                                                97
        DO 11 I=1,NTIC                                                        98
        CALL SYMBOL (XN,YN,0.2*SPACE,13,THETA,-2)                             99
        XN=XN-SPC                                                            100
11      YN=YN-SPS                                                            101
        NTIC=NTIC-1                                                          102
        KOTO=1                                                               103
        IF (ABS(SPS).LT.1.E-03) KOTO=2                                       104
        DO 14 I=1,NTIC                                                       105
        CALL PLOT (XNU,YNU,3)                                                106
        CALL PLOT (XNU,YNU,3)                                                107
        CALL PLOT (XNX,YNY,2)                                                108
        XNX=XNX-SPC                                                          109
        YNY=YNY-SPS                                                          110
        XNU=XNU-SPC                                                          111
        YNU=YNU-SPS                                                          112
        GO TO (12,13),KOTO                                                   113
12      TEMM=XNU                                                             114
        XNU=XNX                                                              115
        XNX=TEMM                                                             116
        GO TO 14                                                             117
13      TEMM=YNU                                                             118
        YNU=YNY                                                              119
        YNY=TEMM                                                             120
14      CONTINUE                                                             121
        RETURN                                                               122
        END                                                                  123
```

APPENDIX E

USER'S MANUAL

HOW TO USE REDUCE II

The input to REDUCE II consists of five basic pieces of information:

(1) A card indicating whether the super G form is ARTIFICIAL or NATURAL

(2) The block diagram inputs

(3) The transfer function (or functions) desired

(4) A set of algebraic expressions representing the block diagram

(5) A sequence of variables separated by commas representing the reduction order string

With the exception of the first card, the input data can be punched anywhere on a card.

Examples of Input to REDUCE II

Examples corresponding to the five basic types of input are described as follows:

(1) This card must have ARTIFICIAL or NATURAL punched on it starting in card column 1.

(2) The block diagram inputs are specified as, for example,

X(7)$X(5)$X(1)$$

A dollar sign separates each input. Two dollar signs in succession (no blanks between them) indicate the end of the block diagram inputs. There may be no more than five inputs for a block diagram as the program is currently dimensioned, and only one card may be used.

(3) The desired transfer function for the block diagram is given by, for example,

X(2)$X(1)$$ or X(2)$$

When two variables are indicated, REDUCE II solves the block diagram for the ratio of the first variable given (the output) to the second (an input). When one variable is given, REDUCE II solves for the ratio of this variable to all the inputs. Again two dollar signs in succession indicate the end of this information, and only one card may be used.

(4) Block diagram expressions are written as

(±N±G(A)*G(B)*...*G(F))*X(B)±G(M)*G(N)*...*X(C)±...$

where A, B, C, ... are integers, N is a numerical constant, the G(i)'s are functions, and the X(i)'s are variables. This allows very general forms that consist of variables X together with their coefficients, which are the sums and products of G's and constants. Notice again that a dollar sign terminates an expression. No more than five cards may be used per expression. An example of an expression card is

$$X(1)+X(2)-X(3)+G(4)*X(9)+(1.+G(1)*G(2))*X(4)\$$$

FORTRAN notation is used to indicate subscripts and arithmetic operations. Also the =0 to make this a mathematical equation is implied. On the very last expression, the user must use two dollar signs in succession. This indicates to REDUCE II that there are no more expression cards to follow.

(5) The reduction order string terminates the input to REDUCE II. Typically, it is

$$X(A),X(B),X(C),,X(D),X(E),,X(F),\ldots X(M)\$$$

where the A, B, C,... are integers. The double commas partition the string into substrings and indicate a super G substitution. The dollar sign terminates the reduction string and should not be immediately preceded by any commas. An example is

$$X(2),X(5),X(6),,X(10),X(7),,X(8)\$$$

There must be no repetition of a variable in the reduction string. All variables but the output and inputs must be represented in the reduction string. Any number of cards may be used. Notice also that one dollar sign terminates the reduction string. These are all the data that REDUCE II needs.

REDUCE II Output

The output consists mainly of a set of cards. Included in these cards are the super G's, the transfer function (or functions), and the transfer function identifiers. EVAL II uses this punched output in several places. The punched deck, after execution of REDUCE II, appears in the sequence shown in figure 17.

REDUCE II prints a copy of the super G cards and the transfer function. Also the user's input in printed. Should REDUCE II detect an error, a message is printed and execution is usually stopped. A list of the messages and their causes follows:

Message	Cause
(1) More than 600 variables	The number of variables exceeds the maximum allowed.

Message	Cause
(2) More than 600 equations	The number of equations exceeds the maximum allowed.
(3) More than 1200 functions	The number of G functions before the reduction starts exceeds the maximum allowed.
(4) More than five block diagram inputs	The number of block diagram inputs exceeds the maximum allowed.
(5) More than five cards for a block diagram expression	Block diagram algebraic expressions may not use more than five cards.
(6) The solution may be in error because equation ___ was not eliminated, equation ___ is	A variable used in the block diagram was not eliminated. This is because it was *not* included in the order string.
(7) Number of variables in order string is not consistent with the number of expressions - execution stopped	There are probably one or more block diagram expressions missing, or there is a mistake in the order string. Check the input data.
(8) Only one equation containing X( )	Only one equation was found containing the variable indicated. This indicates an error in the order string or in an equation. This error message will also be printed if there are no equations with the variable indicated.
(9) Warning - the number of super functions (super G's) exceeds the maximum number of functions (G's) permitted, therefore substitution process stopped	No more super G's will be created since the dimension of G is exceeded. Although the super G substitution process will stop, the reduction process will continue. It is likely that the expressions will become very large and that FORMAC storage space will be exceeded.
(10) Subscript too large for this version	Subroutine INSTRN has found a subscript larger than 600 in the reduction string.
(11) Invalid option specified on super G from card	User has not specified NATURAL or ARTIFICIAL form.

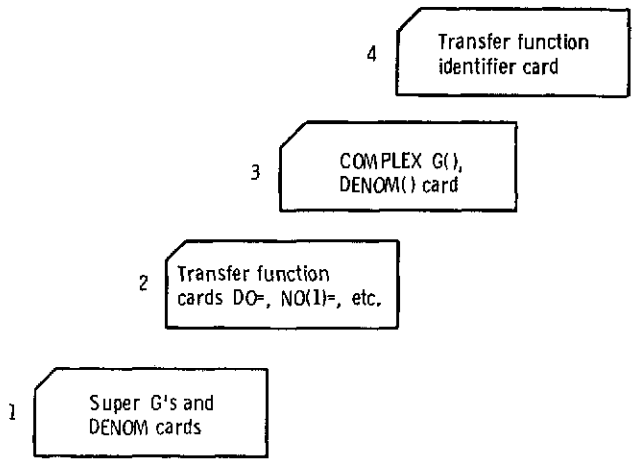


Figure 17. - REDUCE II output card order.

HOW TO USE EVAL II

Before EVAL II can be used, the output from REDUCE II must be rearranged. The FORTRAN-like cards go into subroutine COEFF of EVAL II. The other cards are data to identify transfer functions. They also tell EVAL II how many transfer functions are to be evaluated for any particular run. The cards shown in figure 17 are regrouped as shown in figure 18.

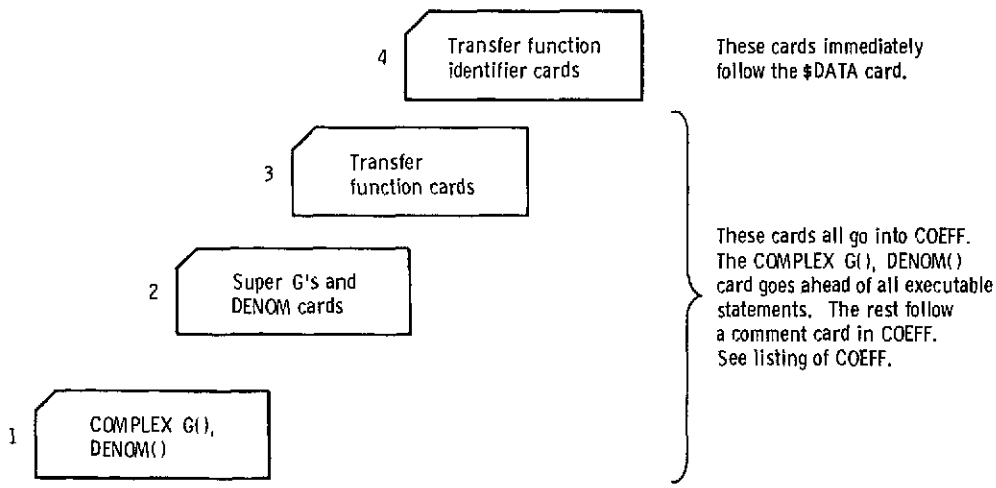


Figure 18. - REDUCE II output regrouped.

The user's functions or G's should be placed before the super G cards. These functions may be of any form as long as they are in FORTRAN. Should they be one of the following forms, the namelist input section of EVAL II can be used:

$$\sin\left(d\sqrt{as^2 + bs + c}\right) = \text{SINHF(I)} \tag{E1}$$

$$\cosh\left(d\sqrt{as^2 + bs + c}\right) = \text{COSHF(I)} \tag{E2}$$

$$\sin\left(d\sqrt{as^2 + bs + c}\right) = \text{SINF(I)} \tag{E3}$$

$$\cos\left(d\sqrt{as^2 + bs + c}\right) = \text{COSF(I)} \tag{E4}$$

$$\exp\left(d\sqrt{as^2 + bs + c}\right) = \text{EXPF(I)} \tag{E5}$$

$$\sqrt{as^2 + bs + c} = \text{SQRTF(I)} \tag{E6}$$

where I is an integer constant.

EVAL II Input

Section I. - The first data cards to EVAL II are the transfer function identifiers that REDUCE II punched out.

Section II. - The next card of the EVAL input must contain $INPUT beginning in card column 2. This starts the namelist section of EVAL II data. The following input forms may appear in any order, but all must begin in card column 2:

(1) NEXP = Number, EXPA = List, EXPB = List, EXPC = List, EXPD = List,
(2) NSIN = Number, SINA = List, SINB = List, SINC = List, SIND = List,
(3) NCOS = Number, COSA = List, COSB = List, COSC = List, COSD = List,
(4) NSINH = Number, SINHA = List, SINHB = List, SINHC = List, SINHD = List,
(5) NCOSH = Number, COSHA = List, COSHB = List, COSHC = List, COSHD = List,
(6) NSQRT = Number, SQRTA = List, SQRTB = List, SQRTC = List,
(7) NK = Number, K = List,
(8) NT = Number, T = List,
(9) NZ = Number, Z = List,
(10) NWN = Number, WN = List,

(11) FSTART = Initial frequency, FEND = Final frequency, DELTAF = Increment (Frequencies are specified in hertz.)

(12) LENGTH = Length of the abscissa in inches.

Any of these forms may use more than one card as long as each card ends with a comma. The suffixes A, B, C, and D in the function names represent the variables implied in the function argument. (Note: no D in SQRT.) For example, EXPA, EXPB, EXPC, and EXPD represent the variables a, b, c, and d in the function

$$\exp\left(d\sqrt{as^2 + bs + c}\right)$$

(See eqs. (E1) to (E6) for additional forms.)

The term "Number," as used in the preceding forms, means an integer number which represents the number of a particular type of function or variable being used. For the functions, this number must not be greater than 5; for variables, not greater than 10. The term "List" means a list of values that the variables are to take on. The elements of this list must be separated by commas and must be in the proper order. For example, NK=2, K=1, 6.25E-2 would set K(1) equal to 1.0 and K(2) equal to 0.0625. For consecutive runs (runs where the transfer function form does not change), only the changed numbers need to be given on a new input. For example,

```
$INPUT
K(2)=1.25E-2,
$
```

A card with a $ in card column 2 must be the very last card of section II. This terminates the namelist data.

Section III (optional). - If LENGTH in the previous section were set to zero, no plots would be made. A positive LENGTH signals EVAL II to make CALCOMP plots. This means that EVAL II needs (1) a plot title and (2) information to make linear or log plots. This information the user punches onto two separate cards.

(1) PLOT TITLE CARD - The user may use all 80 card columns for the plot title.

(2) PLOT TYPE CARD - To get a set of semilogarithmic plots, the word LOGPLOT must appear on this card. LOGPLOT must start in card column 1. A blank card is sufficient to get linear plots.

The user may run several cases of data in one execution of EVAL II. That is, many combinations of system information may be examined in sequence. To do this, the user may change any of the data values in section II. After completing the frequency band FSTART to FEND in steps of DELTAF, EVAL II returns to the statements that read the

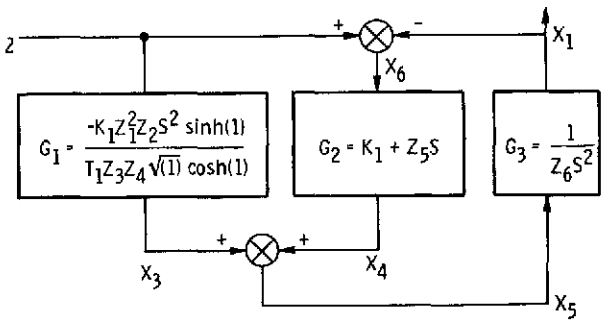
section II (namelist) data. EVAL II would read new data (or stop execution if there is no data) and start anew. Remember that if LENGTH is left greater than zero, a new plot title and new plot type information must be supplied or the program will execute improperly.

EVAL II Output

The output printed by EVAL II lists the following items: (1) All the numerical input; (2) a table to show the correspondence between the transfer function and its identifying number; (3) the frequency band FSTART to FEND in DELTAF; and (4) the identifying number, frequency (Hz), real part, imaginary part, absolute value, phase angle (deg), and magnitude (dB) of the transfer function.

The CALCOMP plots are of two types. The plots on linear axes have absolute value and phase angle as a function of frequency. Those on semilogarithmic axes have decibels as a function of frequency and phase angle as a function of frequency. Each plot is identified with the user-supplied title and the transfer function given as X_A/X_B, where A and B are integers.

A WORKED EXAMPLE



The block diagram shown is quite simple but demonstrates the use of REDUCE II. We want to solve this diagram for X_1 to X_2 using naturally formed super G's. The equations associated with the block diagram are

$$X_3 - G_1X_2 = 0$$

$$X_4 - G_2X_6 = 0$$

$$X_1 - G_3X_5 = 0$$

$$X_2 - X_1 - X_6 = 0$$

$$X_3 + X_4 - X_5 = 0$$

The input data to REDUCE II for this example would be

```
NATURAL FORM
X(2)$$
X(1)$$
X(3)-G(1)*X(2)$
X(2)-X(1)-X(6)$
X(4)-G(2)*X(6)$
X(3)+X(4)-X(5)$
X(1)-G(3)*X(5)$$
X(3),X(4),,X(6),X(5)$
```

The printed output from REDUCE II for this example is

```
THE BLOCK DIAGRAM INPUTS

    X(2)$

THE SOLUTION IS FOR X(1)$  TO THE INPUTS

THE BLOCK DIAGRAM EXPRESSIONS

     1  X(3)-G(1)*X(2)$
     2  X(2)-X(1)-X(6)$
     3  X(4)-G(2)*X(6)$
     4  X(3)+X(4)-X(5)$
     5  X(1)-G(3)*X(5)$$

THE REDUCTION ORDER IS

     1  X(3),X(4),,X(6),X(5)$

THE LARGEST VARIABLE SUBSCRIPT IS     6
THE NUMBER OF EXPRESSIONS IS     5
THE LARGEST FUNCTION (G) SUBSCRIPT IS   3


          * * * OUTPUT FROM REDUCE II * * *

          TRANSFER FUNCTION X(  1)/X(  2)

               THE NUMERATOR
                    G(1)*G(3)+G(2)*G(3)

               THE DENOMINATOR
                    G(2)*G(3)+1.0
```

The cards punched out by REDUCE II are

```
 DC=
*G(2)*G(3)+1.0
 NC( 1)=
*G(1)*G(3)+G(2)*G(3)
 COMPLEX G(    3),DENOM(    1)
1
1
2
```

The FORTRAN expressions for G_1, G_2, and G_3 are

```
G(1)=-K(1)*Z(1)*Z(1)*Z(2)*S*S*SINHF(1)/
     (T(1)*Z(3)*Z(4)*SQRTF(1)*COSHF(1))

G(2)=K(1)+Z(5)*S

G(3)=1./(Z(6)*S*S)
```

These expressions immediately precede the super G output in COEFF. See page 72 for a typical example. The constants associated with this example are

$$K_1 = 7.6336\times10^{5}$$

$$T_1 = 1.0416\times10^{-2}$$

$$Z_1 = 0.75$$

$$Z_2 = 62.4$$

$$Z_3 = 6.48\times10^{7}$$

$$Z_4 = 32.2$$

$$Z_5 = 10$$

$$Z_6 = 1$$

$$\text{SINH(1)} \longleftarrow\!\!\!\!\longrightarrow \text{SINH}\left(d \cdot \sqrt{c + bs + as^2}\right)$$

$$\text{COSH(1)} \longleftarrow\!\!\!\!\longrightarrow \text{COSH}\left(d \cdot \sqrt{c + bs + as^2}\right)$$

$$\sqrt{(1)} \longleftrightarrow \sqrt{c + bs + as^2}$$

where

$$a = 4.1006 \times 10^{-6}$$

$$b = 10^{-4}$$

$$c = 0$$

$$d = 3.333$$

The following cards are the input to EVAL II for this example:

```
    1
    1
    2
 $INPUT
 NSINH=1, SINHA=4.100625E-6,SINHB=1.0E-4,SINHC=0.,SINHD=3.3333,
 NCOSH=1, COSHA=4.100625E-6,COSHB=1.0E-4,COSHC=0.,COSHD=3.3333,
 NSQRT=1,SQRTA=4.100625E-6,SQRTB=1.E-4,SQRTC=0.,
 NK=1,K=.76335938E6,
 NT=1,T=1.0416E-2,
 NZ=6,Z=.750,62.4,6.48E7,32.2,10.,1.,
 FSTART=1.,FEND=50.,DELTAF=1.,LENGTH=10.,
 $
A WORKED EXAMPLE
LINEAR PLOT
```

The output from EVAL II for this example is

FORTRAN EVALUATION OF FORMAC OUTPUT

CALCOMP PLOTS HAVE BEEN REQUESTED

A WORKED EXAMPLE

I	SINHA	SINHB	SINHC	SINHD
1	0.41306E-05	0.10000E-03	0.	0.33333E+01

I	COSHA	COSHB	COSHC	COSHD
1	0.41306E-05	0.10000E-03	0.	0.33333E+01

I	SQRTA	SQRTB	SQRTC
1	0.41306E-05	0.10000E-03	0.

I	K
1	0.76336E+06

I	T
1	0.13416E-01

I	Z
1	0.75000E+00
2	0.62400E+02
3	0.64800E+08
4	0.32200E+02
5	0.10000E+02
6	0.10300E+01

NUMBER 1 TRANSFER FUNCTION X(11/X(2)

FSTART= 1.00000 FEND= 50.00000 DELTAF= 1.00000

NUMBER	FREQUENCY	REAL PART	IMAGINARY PART	ABSOLUTE VALUE	PHASE ANGLE	AMPLITIDE IN DB.
1	1.000000	1.000264	-0.000001	1.000264	-0.000030	0.002296
1	2.000000	1.001059	-0.000004	1.001059	-0.000238	0.009195
1	3.000000	1.002390	-0.000014	1.002390	-0.000807	0.020730
1	4.000000	1.004264	-0.000034	1.004264	-0.001930	0.036958
1	5.000000	1.006695	-0.000067	1.006695	-0.003810	0.057957
1	6.000000	1.009698	-0.000118	1.009698	-0.006671	0.083832
1	7.000000	1.013295	-0.000190	1.013295	-0.010762	0.114717
1	8.000000	1.017510	-0.000291	1.017510	-0.016364	0.150772
1	9.000000	1.022374	-0.000425	1.022374	-0.023798	0.192196
1	10.000000	1.027924	-0.000600	1.027924	-0.033437	0.239220
1	11.000000	1.034204	-0.000825	1.034204	-0.045719	0.292123
1	12.000000	1.041265	-0.001112	1.041266	-0.061162	0.351231
1	13.000000	1.049170	-0.001472	1.049172	-0.080388	0.416930
1	14.000000	1.057993	-0.001923	1.057995	-0.104148	0.489671
1	15.000000	1.067821	-0.002485	1.067824	-0.133356	0.569991
1	16.000000	1.078758	-0.003185	1.078763	-0.169141	0.658519
1	17.000000	1.090931	-0.004054	1.090938	-0.212902	0.756004
1	18.000000	1.104491	-0.005135	1.104503	-0.266397	0.863239
1	19.000000	1.119624	-0.006485	1.119643	-0.331851	0.981593
1	20.000000	1.136558	-0.008175	1.136587	-0.412119	1.112056
1	21.000000	1.155573	-0.010304	1.155619	-0.510904	1.256293
1	22.000000	1.177022	-0.013006	1.177094	-0.633071	1.416223
1	23.000000	1.201352	-0.016463	1.201465	-0.785108	1.594220
1	24.000000	1.229135	-0.020935	1.229313	-0.975805	1.793248
1	25.000000	1.261115	-0.026797	1.261400	-1.217290	2.017056
1	26.000000	1.298278	-0.034601	1.298739	-1.526637	2.270440
1	27.000000	1.341946	-0.045184	1.342707	-1.928449	2.559623
1	28.000000	1.393926	-0.059864	1.395211	-2.459133	2.892795
1	29.000000	1.456726	-0.080787	1.458964	-3.174252	3.280894
1	30.000000	1.533872	-0.111610	1.537927	-4.161703	3.738714
1	31.000000	1.630300	-0.158890	1.638024	-5.566511	4.286406
1	32.000000	1.752605	-0.235092	1.768303	-7.639974	4.951131
1	33.000000	1.907810	-0.365391	1.942485	-10.842202	5.767153
1	34.000000	2.094135	-0.602963	2.179212	-16.062666	6.765591
1	35.000000	2.254446	-1.053980	2.488654	-25.056649	7.919291
1	36.000000	2.110127	-1.819422	2.786204	-40.769007	8.900259
1	37.000000	1.178423	-2.424559	2.695768	-64.078579	8.613651
1	38.000000	0.153711	-2.014335	2.020192	-85.636287	6.107851
1	39.000000	-0.136001	-1.303576	1.310651	-95.956075	2.349741
1	40.000000	-0.071503	-0.836708	0.839758	-94.884479	-1.516920
1	41.000000	0.061374	-0.569645	0.572942	-83.850584	-4.837784
1	42.000000	0.186983	-0.412119	0.452553	-65.595722	-6.886604
1	43.000000	0.292803	-0.313532	0.428994	-46.958082	-7.350979
1	44.000000	0.380371	-0.248257	0.454217	-33.131272	-6.854729
1	45.000000	0.453432	-0.202922	0.496768	-24.109711	-6.076932
1	46.000000	0.515370	-0.170171	0.542738	-18.272824	-5.308194
1	47.000000	0.568820	-0.145730	0.587191	-14.369910	-4.624406
1	48.000000	0.615761	-0.126993	0.628721	-11.653161	-4.030847
1	49.000000	0.657672	-0.112305	0.667192	-9.690418	-3.514981
1	50.000000	0.695668	-0.100572	0.702900	-8.226227	-3.062131

PLOT MADE

PLOT MADE

REFERENCES

1. Lorenzo, Carl F.; and Swigert, Paul: Computer Program for Symbolic Reduction of Block Diagrams Using FORMAC. NASA TN D-4617, 1968.

2. Pavkovich, John M.: The Solution of Large Systems of Algebraic Equations. Tech. Rep. 33, Stanford Univ. (AD-427753), Dec. 6, 1963.

3. Bond, E. R.: User's Preliminary Reference Manual for FORMAC. Boston Advanced Programming Dept., International Business Machines, Feb. 1964.

4. Busacker, Robert G.; and Saaty, Thomas L.: Finite Graphs and Networks, An Introduction With Applications. McGraw-Hill Book Co., Inc., 1965.

5. Seshu, Sundaram; and Reed, Myril B.: Linear Graphs and Electrical Networks. Addison-Wesley Publ. Co., 1961.

6. Sammet, Jean E.: Formula Manipulation by Computer. Advances in Computers. Vol. 8. F. L. Alt and Morris Rubinoff, eds. Academic Press, 1967, pp. 47-102.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1974—739-160/128