

DACS: AN INTERACTIVE COMPUTER PROGRAM
TO AID IN THE DESIGN AND ANALYSIS OF
LINEAR CONTROL SYSTEMS

by

W. Ivan Lewis, III

Thesis submitted to the Graduate Faculty of the
Virginia Polytechnic Institute and State University in
partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

APPROVED:

A. Myklebust, Chairman

H. H. Robertshaw

S. W. Zewari

R. G. Leonard

March 1985

Blacksburg, Virginia

(C) Copyright by W. Ivan Lewis, III 1985

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to the following people for their help and guidance.

Dr. Arvid Myklebust, for serving as my major professor and for lending his support and encouragement throughout this project.

Dr. Harry Robertshaw, for serving on my committee, for exercising DACS to its limits, and for the inspiration he provided.

Dr. Robert Leonard, for serving on my committee, for helping with the DACS manual and this thesis, and for offering advice whenever needed.

Dr. Said Zewari, for serving on my committee, and offering his comments on how to make software easy to use.

Mr. , Mr. , Mr. , and Mr.

whose friendships made my studies at Virginia Tech enjoyable.

My parents, and Mrs. who provided the love, patience, support and encouragement which made my entire college education possible.

Mrs. for welcoming me into her home and providing love and friendship over the past several years.

The family for providing a home away from home and for their love and encouragement.

Finally, I would like to thank my wife-to-be, , whose love has changed my life in so many ways.

TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGMENTS	iii
LIST OF FIGURES	vi
1. INTRODUCTION	1
2. PROGRAM DESCRIPTION AND STRUCTURE	3
3. PROGRAM FEATURES	7
4. PROGRAM THEORY	10
4.1 Root Locus	10
4.2 Frequency Response	13
4.3 System Simulation	20
4.4 Time Response	23
4.5 Matrix Functions	26
5. A PROGRAM SESSION	29
6. DESIGN EXAMPLES	49
6.1 Servo Positioning System	49
6.2 Electro-Hydraulic Ship Steering System	61
7. CONCLUSIONS AND RECOMMENDATIONS	67
REFERENCES	69
APPENDIX A: DACS Calculation Examples	70
A.1 Root Locus	71
A.2 Frequency Response	72
A.3 System Simulation	73
A.4 Time Response	80
A.5 Matrix Functions	82

TABLE OF CONTENTS (continued)

	<u>Page</u>
APPENDIX B: DACS User's Guide	98
APPENDIX C: DACS Program Listings	129
C.1 DACS	130
C.2 ROOT	136
C.3 FREQ	156
C.4 SIM	182
C.5 RAT	205
C.6 MAT	226
VITA	241
ABSTRACT	

LIST OF FIGURES

	<u>Page</u>
1. DACS Structure	4
2. DACS Program Module Control Structure	6
3. General System Block Diagram	11
4. Nichols Chart Showing Bandwidth	17
5. Displaced Nichols Chart Showing Bandwidth	19
6. DACS Opening Message	30
7. DACS Main Menu with Default Settings	31
8. Root Locus Menu	33
9. Help Screen on Root Locus Model Entry	34
10. Help Screen on Root Locus Parameter Input	35
11. Root Locus Graphing Options	37
12. Example System Root Locus	38
13. Example System Root Locus Showing Cursor Option	40
14. Root Locus Data Menu	41
15. Example System Roots for $K = 0$	42
16. Root Locus File Help Screen	43
17. Frequency Response Menu	45
18. System Simulation Menu	46
19. Time Response Menu	47
20. Matrix Functions Menu	48
21. Servo System	50
22. Servo System Block Diagram	51
23. Reduced Servo Diagram	52
24. Servo Root Locus ($K = 1 - 20$)	54

LIST OF FIGURES (continued)

	<u>Page</u>
25. Servo Step Response ($K = 1$)	55
26. Servo Step Response ($K = 4$)	56
27. Servo Step Response ($K = 5$)	57
28. Servo Step Response ($K = 10$)	58
29. Servo Step Response ($K = 20$)	59
30. Servo Step Response with Disturbance	60
31. Electro-Hydraulic Ship Steering System	62
32. Steering System Block Diagram	64
33. Steering System Response	66
34. Root Locus for Equation 52	72
35. Bode Amplitude and Phase Example	74
36. Bode Magnitude Example	75
37. Bode Phase Example	76
38. Nyquist Diagram Example	77
39. Nichols Chart Example	78
40. System Simulation Example	81
41. Time Response Example - First Vector	83
42. Time Response Example - Second Vector	84
43. Time Response Example - Third Vector	85
44. Time Response Example - Fourth Vector	86
45. Time Response Example - Output Equation	87
46. Matrix Determinant Example	89
47. Matrix Inverse Example	90

LIST OF FIGURES (continued)

	<u>Page</u>
48. Matrix Characteristic Polynomial Example	91
49. Matrix Eigenvalues Example	92
50. Matrix State Transition Matrix Example ($T = 1$)	93
51. Matrix Resolvent Example - First Coefficient	94
52. Matrix Resolvent Example - Second Coefficient	95
53. Matrix Resolvent Example - Third Coefficient	96
54. Matrix Resolvent Example - Fourth Coefficient	97
55. Example Root Locus	123
56. Zoom on Example Root Locus	125

1.0 INTRODUCTION

Several computer programs are currently available to aid in control system design and evaluation. Notable among these are CSMP [1], ACSL [2], Super-Sceptre [3], and DSL [4]. These programs are designed to run on large computing systems providing control system information primarily through simulation.

Recent advances in computer hardware technology have brought a significant amount of computational ability to the desktop computer. This coupled with several dozen universities across the country requiring their engineering students to own personal computers [5] is producing a new breed of engineer who will rely heavily on the desktop personal computer to aid in engineering problem solving. This trend has created a growing need for engineering software for personal computers.

The purpose of this thesis is to help meet this software need in the area of linear control systems. Specifically, this thesis presents an interactive computer program developed for the IBM Personal Computer. This program, called DACS (Design and Analysis of Control Systems), allows the user to investigate linear control systems through graphical, numerical and analytical means. Written in compiled basic, DACS includes the capability to generate Root Locus, Bode, Nyquist, and Nichols diagrams. It can also provide system simulation and closed-form time response calculations. In addition, the user may obtain matrix information such as determinants, inverses, characteristic equations, eigenvalues, state transition matrices and resolvent matrices. Calculated data may be displayed in either graphical or tabular form.

Every effort was made to develop an easy-to-use yet powerful linear system design tool.

2.0 PROGRAM DESCRIPTION AND STRUCTURE

DACS is designed to run on the IBM Personal Computer, IBM Personal Computer Portable, IBM Personal Computer AT, or compatible computers. DACS is written entirely in compiled basic, is menu driven, and requires 160kilobytes of memory, one disk drive and a color-graphics monitor. DACS consists of five stand-alone programs:

1. Root Locus
2. Frequency Response
3. System Simulation
4. Time Response
5. Matrix Functions

The Root Locus module provides a plot of the location of the roots of a system as a function of some parameter, usually gain. The Frequency Response module provides Bode plots, Nichols charts, and Nyquist diagrams of a system. The System Simulation module numerically simulates system response to a given input, while the Time Response module calculates the time response of a system in closed-form. The Matrix Functions module is a collection of matrix manipulation programs that calculate determinants, inverses, characteristic polynomials, eigenvalues, state transition matrices, and resolvent matrices. A state-space model form is used to describe the system in the Simulation and Time Response modules while a system transfer function is used in the Root Locus and Frequency Response modules.

Each of the program modules may be selected from the main menu as depicted in Fig. 1. DACS was purposely divided into modules to reduce

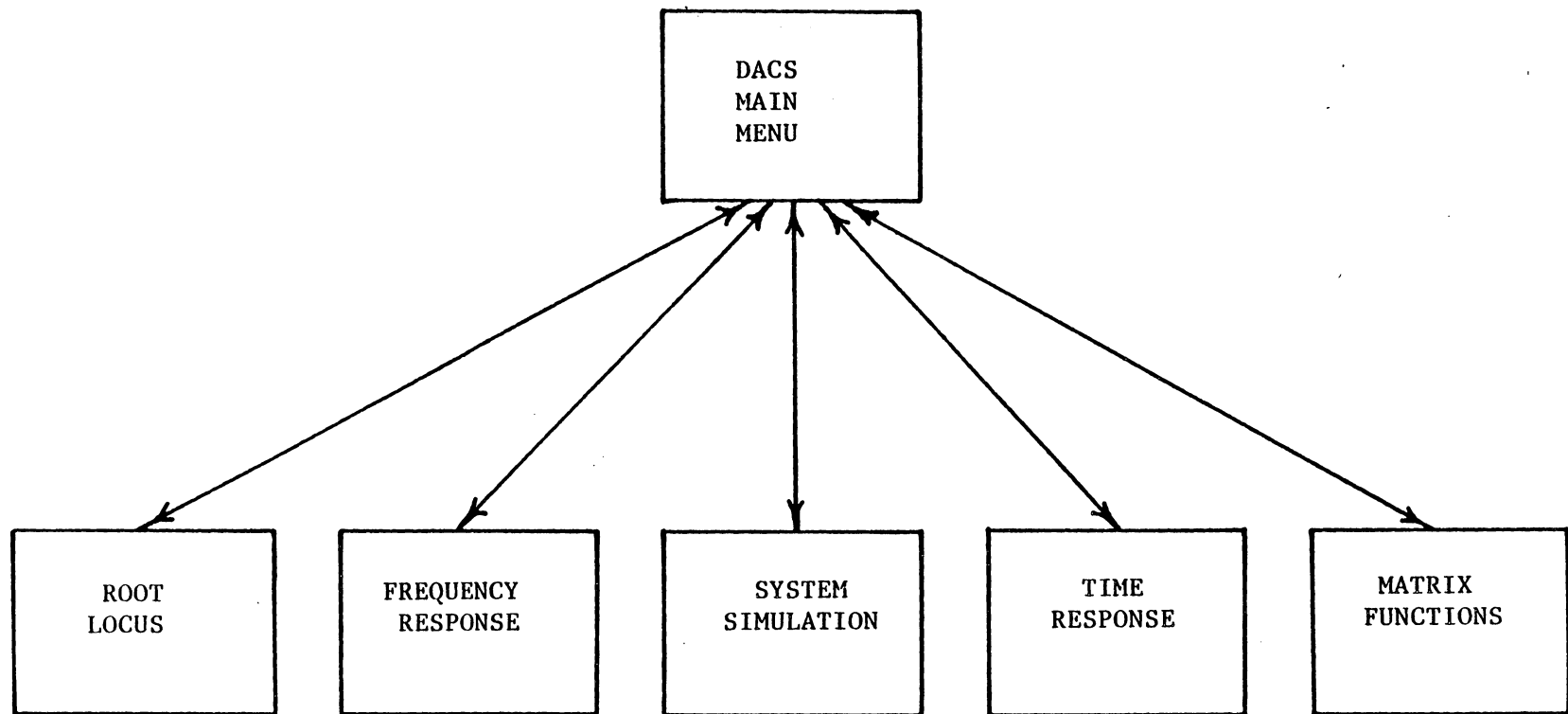


Figure 1. DACS Structure

the amount of required user input. For example, a user will typically perform several frequency response calculations in sequence before moving to another type of analysis. Once the user has moved into the Frequency Response module the program knows that all models will be frequency response models, all calculations will be frequency response calculations, and so forth. In this way, the user is not required to instruct the program as to what type of calculations are desired each time. When another type of analysis is to be performed, the user simply returns to the main menu and selects another module.

The control structure of each individual module is basically the same as shown in Fig. 2. To enter a model for analysis, the user selects the ENTER MODEL option from the local menu. After entering the model, the user is returned to the local menu so that another option may be selected. User input is purposely distributed under several menu selections to minimize input. Here, for instance, the user may calculate data on the same model for several different inputs without having to re-enter the model each time.

Figure 2 also shows several of the features available in each program module. For example, models may be entered from the console, saved on disk and recalled from disk. Calculated data may be graphed or listed. A graph may be displayed on a monitor, sent to a printer, saved on disk, and recalled from disk. Data may be listed on the monitor, saved on disk, and recalled from disk. All of these options serve to enhance user interaction with DACS.

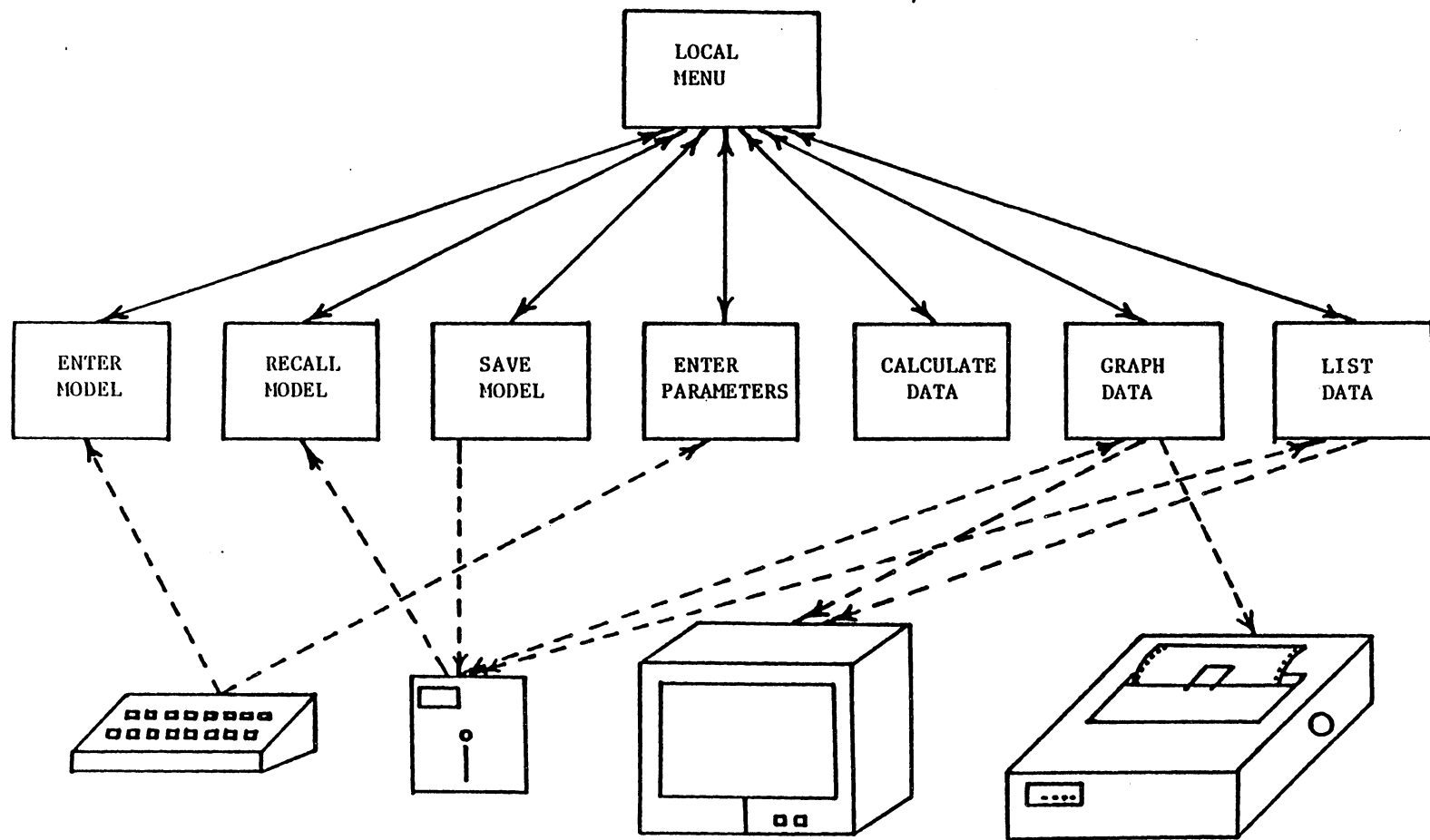


Figure 2. DACS Program Module Control Structure

3.0 PROGRAM FEATURES

This section describes general program features added to aid in program use. These features are applicable to all program modules.

Among the features added are:

1. Function Key Input: All menu selections make use of the IBM Personal Computer's ten function keys.
2. HELP Option: HELP may be toggled on and off from any menu. When HELP is on, all required user input is explained in detail on the monitor. HELP may be turned off to expedite parameter input.
3. Sound: Three levels of sound are provided. Sound level "two" provides user input signals and error signals for incorrect input. Sound level "one" provides error signals only, while sound level "zero" disables sound completely. The sound level may be changed from any menu and the current sound level is displayed on each menu.
4. Session Archiving: When the archiving option is activated an archive file is kept by the program. In this file is stored a record of all menu selections chosen and all filenames used. In addition, comments may be entered into the archive at any time. This file provides a complete record of the user's session with DACS and may be read with a DOS command.
5. Saving/Recalling/Modifying User Input: All models may be saved on disk and recalled from disk. In addition, all models may be modified on an element-by-element basis to eliminate

the need to re-enter a model if only a minor change is desired.

6. Saving/Recalling Calculated Data: All of the data calculated by DACS may be saved on disk and recalled later. This allows the user the option of regraphing and re-analyzing previously calculated data.
7. Disk File Information: The user may obtain a listing of the files on a disk from any local menu. In addition, each time data is saved on or recalled from disk, both file directory and filename information may be obtained. Unwanted files may be erased from within the application.
8. Escape Capability: The escape key provides an exit to a higher command level at all times.
9. Default Answers: There are several times when DACS will ask a yes-or-no question. The default answer is always "no". Pressing the carriage return is equivalent to a no reply.
10. Erasement Checks: There are several checks built into DACS to insure that information is not lost. For example, if a model has already been entered into the program and the ENTER MODEL option is selected, DACS will ask if the existing model is to be erased. A yes answer is required before any action is taken.

Once models have been entered and calculations have been made, DACS allows the user a graphical or tabular display of the data. Some of the graphical features of DACS include:

1. Automatic Scaling: All graphs are originally scaled by DACS for the user.
2. Zooming: Once a graph is displayed by DACS, the user may zoom in or out on the graph as often as desired.
3. Graph Interaction: When a graph is displayed the user may move the cursor to any point and get graph coordinate information for that point.
4. Screen Dumps: Any displayed graph may be sent to a graphics printer or saved in a file on disk and redisplayed at a later time.
5. Grids: Graphs may be displayed with or without grids.

4.0 PROGRAM THEORY

This section describes the calculation method used in each program module. The highest order system that each particular module can accommodate is given in each subsection. In all cases this limit is imposed by compiler memory limitations. There are no order limits on the algorithms used.

4.1 Root Locus

The Root Locus module may be used to obtain a plot of the location of the roots of the characteristic equation or eigenvalues of a system as a function of some independent parameter, usually gain. The system is described to the program by a transfer function of the form

$$GH(s) = K \frac{N(s)}{D(s)} \quad (1)$$

where K is the independent parameter. Here $GH(s)$ represents the transfer function of the system shown in Fig. 3. This system has the closed-loop characteristic equation

$$1 + K \frac{N(s)}{D(s)} = 0 \quad (2)$$

since there is negative feedback. For the case of studying the effect of a general independent parameter K , the actual system's characteristic equation may be manipulated into the form of equation 2.

Equation 2 is equivalent to

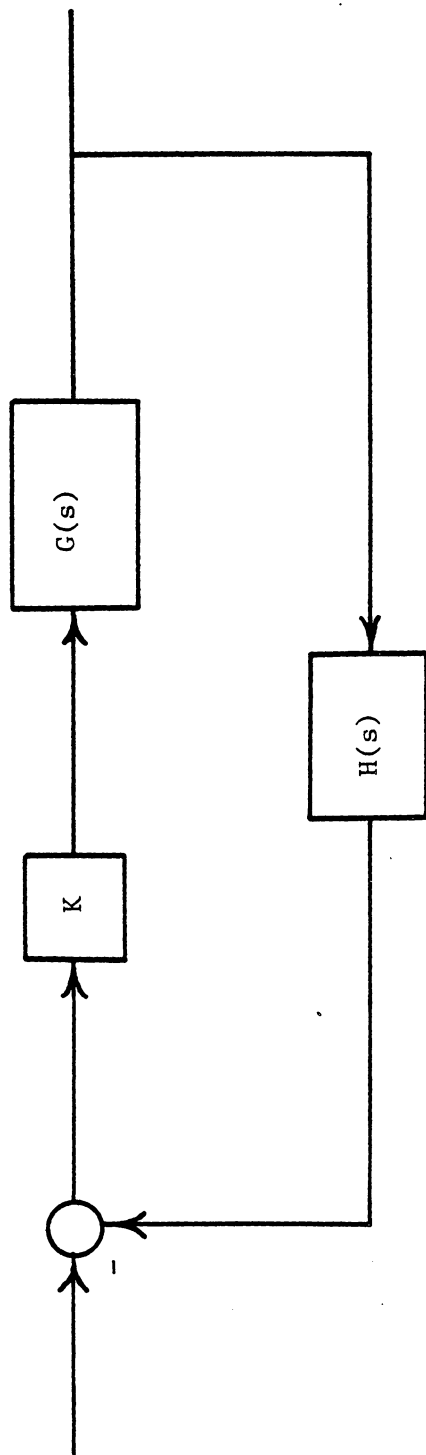


Figure 3. General System Block Diagram

$$D(s) + K N(s) = 0 \quad (3)$$

The numerator and denominator polynomials of the system transfer function are written as

$$N(s) = s^p + n_p s^{p-1} + \dots + n_2 s + n_1 \quad (4)$$

and

$$D(s) = s^q + d_q s^{q-1} + \dots + d_2 s + d_1 \quad (5)$$

The root locus of the system described by Eq. 1 is now obtained by factoring Eq. 3 for each value of K.

For systems of degree five or larger, Bairstow's method is used to find the roots of the system characteristic equation. Bairstow's method is an iterative method which involves finding quadratic factors of the polynomial. Once found, two roots of the polynomial may be obtained from the quadratic factor by using the familiar quadratic equation. Two more roots may then be obtained by repeating the process on the $n-2$ degree polynomial resulting from the extraction of the quadratic factor for the previous two roots. The procedure is continued until all the roots of the original polynomial are found.

For systems of degree less than five, the roots of the system characteristic equations are found in complete closed-form.

The Root Locus module requires from the user the polynomial coefficients n_i and d_i , the range of K of interest and the incremental step for K . The program then forms the system characteristic equation and uses the appropriate root finding method to find the roots of the characteristic equation for each K . Systems of up to twelfth order may be analyzed.

4.2 Frequency Response

The Frequency Response module may be used to obtain a Bode plot, Nyquist diagram, or Nichols chart of a system. As with the Root Locus module, the system is described to the program by a transfer function of the form given in Eqs. 1, 4, and 5. In this case, $GH(s)$ represents the system open-loop transfer function and K represents the system gain and does not vary.

For Bode plots and Nichols charts, the numerator and denominator polynomials may first be factored to yield a transfer function in the form

$$GH(s) = K \frac{(s+a_1+b_1j)(s+a_2+b_2j) \dots (s+a_p+b_pj)}{(s+c_1+d_1j)(s+c_2+d_2j) \dots (s+c_q+d_qj)} \quad (6)$$

In this equation a_i and b_i represent the real and imaginary parts of each root of the numerator polynomial and c_i and d_i represent the same parameters for the denominator polynomial. The frequency transfer function corresponding to Eq. 6 is obtained by replacing s with $j\omega$ to yield

$$GH(j\omega) = K \frac{(j\omega + a_1 + b_1 j)(j\omega + a_2 + b_2 j) \dots (j\omega + a_p + b_p j)}{(j\omega + c_1 + d_1 j)(j\omega + c_2 + d_2 j) \dots (j\omega + c_q + d_q j)} \quad (7)$$

The amplitude ratio of this function can be written as

$$AR(\omega) = K \frac{((\omega + b_1)^2 + a_1^2)^{1/2} \dots ((\omega + b_p)^2 + a_p^2)^{1/2}}{((\omega + d_1)^2 + c_1^2)^{1/2} \dots ((\omega + d_q)^2 + c_q^2)^{1/2}} \quad (8)$$

If the amplitude ratio is expressed in terms of decibels then Eq. 8 becomes

$$\begin{aligned} AR_{db} = & 20 \log K + 20 \log ((\omega + b_1)^2 + a_1^2)^{1/2} + \dots \\ & + 20 \log ((\omega + b_p)^2 + a_p^2)^{1/2} \\ & - 20 \log ((\omega + d_1)^2 + c_1^2)^{1/2} - \dots \\ & - 20 \log ((\omega + d_q)^2 + c_q^2)^{1/2} \end{aligned} \quad (9)$$

Thus, the amplitude ratio in decibels is merely the sum of the individual amplitude ratio of each factor expressed in decibels.

In a similar manner, the phase angle of the frequency transfer function may be found by summing the angles associated with each term. A minus sign is associated with angles in the denominator.

For Nyquist diagrams the frequency transfer function may be written in unfactored form as

$$GH(j\omega) = K \frac{(j\omega)^p + n_p (j\omega)^{p-1} + \dots + n_2 (j\omega) + n_1}{(j\omega)^q + d_q (j\omega)^{q-1} + \dots + d_2 (j\omega) + d_1} \quad (10)$$

By performing complex arithmetic the transfer function may be evaluated at a specified frequency to yield the real and imaginary parts of $GH(j\omega)$.

The Frequency Response module requires from the user the polynomial coefficients n_1 and d_1 , the system gain K , the range of frequency of interest, and the number of frequencies for which calculations are to be performed. The program then prompts the user as to whether Bode, Nyquist, or Nichols calculations are desired. If Bode calculations are chosen then, in order to obtain true phase information, the program factors the numerator and denominator polynomials of the system transfer function using the same factoring routines employed in the Root Locus module. For each value of frequency determined by a logarithmic interpolation between the minimum and maximum values specified by the user, the amplitude and phase are calculated as explained above. Two curves are generated, magnitude versus frequency and phase versus frequency. These two plots may be viewed simultaneously on the screen display or individually.

If a Nichols chart is desired then calculation proceeds exactly as with Bode calculations except frequency is interpolated linearly between the minimum and maximum specified values. The resulting curve for Nichols calculations is a plot of magnitude versus phase and the frequency variable is eliminated. When a Nichols chart is displayed, however, the user may move the cursor to a point on the curve and obtain open-loop magnitude and phase, closed-loop magnitude and phase, and frequency information about that point. The open-loop magnitude and

phase of the point are determined from the calculated data. The closed-loop information is obtained from the calculated open-loop data by the two formulas

$$M = \frac{G}{(G^2 + 2G \cos \theta + 1)^{1/2}} \quad (11)$$

and

$$\tan \phi = \frac{\sin \theta}{G + \cos \theta} \quad (12)$$

where

M = closed-loop magnitude

G = open-loop magnitude

ϕ = closed-loop phase

θ = open-loop phase

(Note: The Nichols calculations use unity feedback for closed-loop information.) Frequency information is obtained by a table search of the calculated data.

On each Nichols chart is drawn the open-loop 0 decibel line and the -180° phase line as well as the closed-loop -3 decibel contour. These represent important reference points for control system design. For example, consider the Nichols chart of Fig. 4. The gain margin is, by definition, the negative of the open-loop magnitude where the system curve crosses the -180° phase line. For the system in Fig. 4, the gain

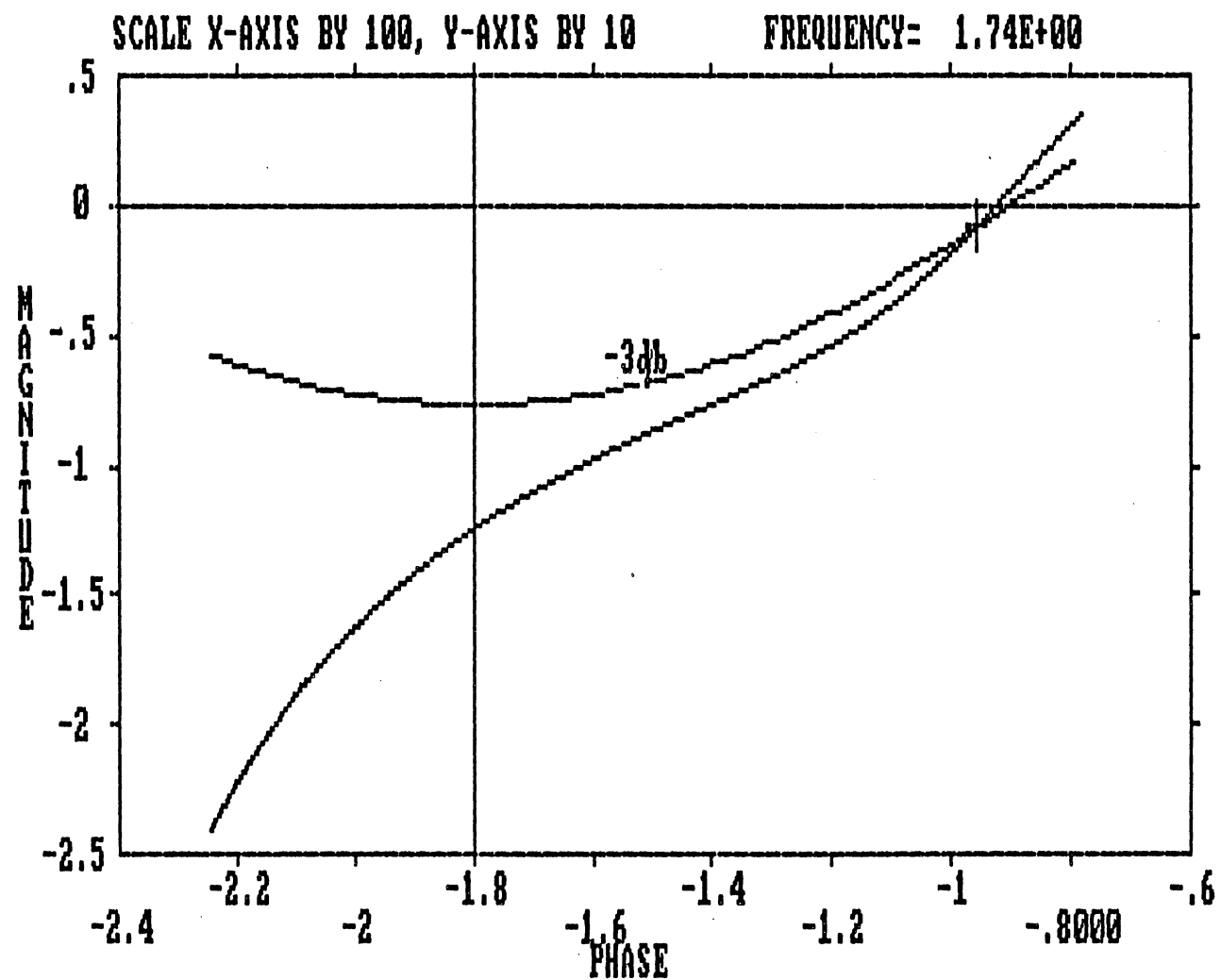


Figure 4. Nichols Chart Showing Bandwidth

margin is 12.5 decibels. The phase margin is defined as 180° plus the open-loop phase angle where the system curve crosses the 0 decibel line. For the system of Fig. 4, the phase margin is 87.3° . The bandwidth may be determined from the intersection of the system curve and the closed-loop -3 decibel contour. As can be seen in Fig. 4, the cursor was moved to the intersection and the frequency at that point was found to be 1.74 radians/second which is the system bandwidth. Therefore, the Nichols chart provides the user with open-loop magnitude and phase, closed-loop magnitude and phase, phase margin, gain margin and bandwidth information all on the same plot.

Note that if the system gain were to be changed the Nichols chart would remain the same except for a vertical displacement. Therefore, if the curve was displaced vertically, the effect of gain on several parameters including gain margin, phase margin and bandwidth could be found. DACS allows the user to perform such a displacement. As an example, it was desired to check the effect of a 10 decibel increase in gain on the system of Fig. 4. By specifying this 10 decibel offset to DACS, the curve of Fig. 5 was generated. By moving the cursor to various points of interest it was found that the new gain margin was 2.5 decibels, the new phase margin was 17° and the new bandwidth was 7.68 radians/second. Thus, increasing the gain decreased system stability and increased the system bandwidth.

If Nyquist calculations are performed then the unfactored frequency transfer function, Eq. 10, is evaluated at each frequency. The result is a plot of the imaginary part versus the real part of the complex

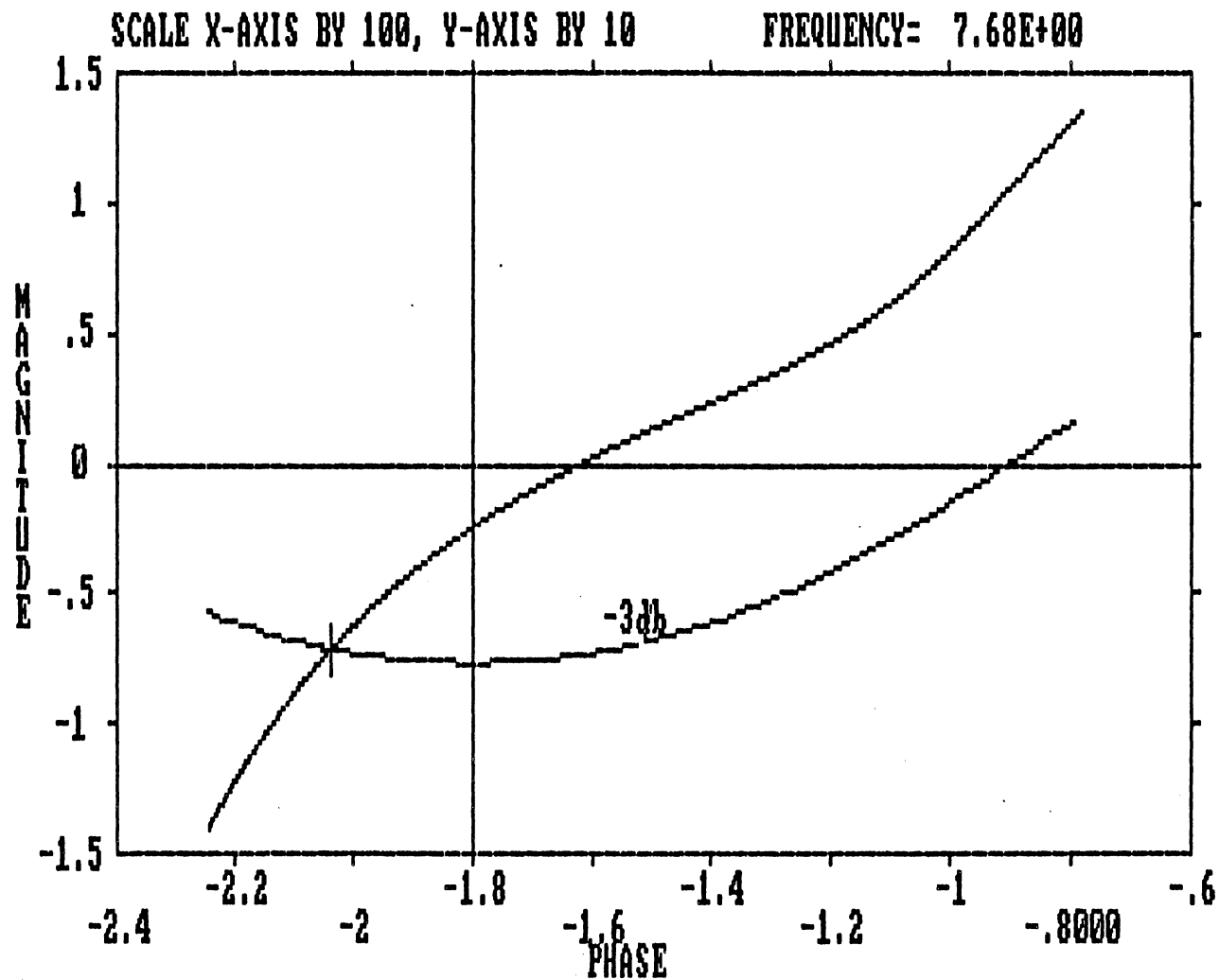


Figure 5. Displaced Nichols Chart Showing Bandwidth

function. Here, frequency is linearly interpolated between the limits specified by the user.

The Frequency Response module can analyze systems of twentieth order or less.

4.3 System Simulation

The System Simulation module may be used to numerically simulate the response of a system to a given input. The system is described to the program by a state model of the form

$$\dot{\underline{x}}(t) = \underline{A}\underline{x}(t) + \underline{B}\underline{u}(t) \quad (13)$$

$$\underline{y}(t) = \underline{C}\underline{x}(t) + \underline{D}\underline{u}(t) \quad (14)$$

where

$\underline{x}(t)$ = vector of system states

$\underline{u}(t)$ = vector of system inputs

$\underline{y}(t)$ = output vector

and A, B, C, and D are constant matrices. The solution to this set of equations is made up of two terms, the homogeneous solution and the particular integral. The solution appears as

$$\underline{x}(t_j) = e^{\underline{A}(t_j - t_1)} \underline{x}(t_1) + \int_{t_1}^{t_j} e^{\underline{A}(t_j - \tau)} \underline{B} \underline{u}(\tau) d\tau \quad (15)$$

where t_j is some time greater than t_1 .

Consider time to be broken down into discrete intervals all with the same length T, that is $t_{i+1} - t_i = T$, then the solution may be

written as

$$\underline{x}((n+1)T) = e^{AT} \underline{x}(T) + \int_{nT}^{(n+1)T} e^{A((n+1)T-\tau)} B \underline{u}(\tau) d\tau \quad (16)$$

If the input, $\underline{u}(\tau)$ is assumed constant over the interval of integration, that is $\underline{u}(\tau) = \underline{u}(nT)$ for $nT \leq \tau < (n+1)T$, then the input may be removed from within the integral resulting in

$$\underline{x}((n+1)T) = e^{AT} \underline{x}(T) + \int_{nT}^{(n+1)T} e^{A((n+1)T-\tau)} d\tau B \underline{u}(nT) \quad (17)$$

The matrix exponential e^{AT} , the state transition matrix, can be calculated by the infinite series

$$\phi(T) = e^{AT} = I + AT + \frac{1}{2!} A^2 T^2 + \frac{1}{3!} A^3 T^3 + \dots \quad (18)$$

The key to the calculation of Eq. 18 then becomes determining the integral

$$\int_{nT}^{(n+1)T} e^{A((n+1)T-\tau)} d\tau \quad (19)$$

Using properties of the state transition matrix and variable substitution, it can be shown that this integral is equivalent to

$$\int_0^T e^{At} dt \quad (20)$$

Substituting the infinite series for the state transition matrix into the above integral and integrating term by term results in

$$\phi(T) = \int_0^T e^{At} dt = IT + \frac{1}{2!} AT^2 + \frac{1}{3!} A^2 T^3 + \dots \quad (21)$$

Equation 17 now becomes

$$\underline{x}((n+1)T) = \phi(T)\underline{x}(nT) + \phi(T) B\underline{u}(nT) \quad (22)$$

Thus, knowing the state of a system and the input to that system at some time nT , the state of the system at time $(n+1)T$ may be calculated.

The simulation module requires from the user the system matrices A , B , C , and D , the time interval T , the input to the system $\underline{u}(t)$, the initial state of the system $\underline{x}(0)$, and the length of simulation. The program calculates the matrix $\phi(T)$ by truncating the infinite series of Eq. 21. The series is truncated when the addition of terms results in less than a one percent change in $\phi(T)$. The state transition matrix is calculated next by the equation

$$\phi(T) = I + A\phi(T) \quad (23)$$

Knowing $\phi(T)$, $\phi(T)$, $\underline{x}(0)$ and the system input, Eq. 22 may be used to calculate the system state at time T . This new information may then be used to calculate the state at time $2T$ and so on until the length of simulation has been satisfied.

The only approximation in the solution method is that the input is constant over the time interval. Therefore the user must pick an appropriate interval, taking into consideration the input and the system time constants.

The simulation module can handle twentieth order systems with up to twenty inputs. Each input may be assigned to be null, a step, a ramp, or a sine wave. Up to five outputs may be plotted as a function of time or as a function of another output.

4.4 Time Response

The Time Response module may be used to calculate the closed-form time response of a system described by a state model as

$$\dot{\underline{x}}(t) = \underline{A}\underline{x}(t) + \underline{b}u(t) \quad (24)$$

$$u(t) = K[r(t) - \underline{k}^T \underline{x}(t)] \quad (25)$$

$$y(t) = \underline{c}^T \underline{x}(t) \quad (26)$$

where

$\underline{x}(t)$ = vector of system states

$u(t)$ = single input

$y(t)$ = single output

$\underline{A}$, $\underline{b}$, $\underline{c}$, and $\underline{k}$ are system matrices and vectors and K is the input gain. The input $u(t)$ is a combination of a feedback of system states and an externally applied input $r(t)$. The externally applied input is described to the program as a transfer function of the form

$$R(s) = K_r \frac{N(s)}{D(s)} \quad (27)$$

The solution method is to combine the input with the system to form a homogeneous augmented system of the form

$$\dot{\underline{x}}_a(t) = A_a \underline{x}_a(t) \quad (28)$$

The solution to this homogeneous system then becomes

$$\underline{x}_a(t) = e^{A_a t} \underline{x}_a(0) \quad (29)$$

Substituting Eq. 25 for $u(t)$ into Eq. 24 and collecting terms results in the following expression for $\dot{\underline{x}}(t)$

$$\dot{\underline{x}}(t) = [A - K\underline{b}k^T] \underline{x}(t) + K\underline{b}r(t) \quad (30)$$

By using phase variables, the external input $r(t)$ can be represented as a state model in the form

$$\dot{\underline{x}}_r(t) = A_r \underline{x}_r(t) \quad (31)$$

$$y_r(t) = \underline{c}_r^T \underline{x}_r(t) \quad (32)$$

where $y_r(t)$ has the identical time response of $r(t)$. This identical time response condition allows Eq. 32 to be substituted for $r(t)$ in Eq.

30 resulting in

$$\dot{\underline{x}}(t) = [A - K\underline{b}\underline{k}^T] \underline{x}(t) + K\underline{b}\underline{c}_r^T(t) \underline{x}_r(t) \quad (33)$$

This equation and Eq. 31 now completely define the homogeneous augmented system as

$$\begin{bmatrix} \dot{\underline{x}}(t) \\ \vdots \\ \dot{\underline{x}}_r(t) \end{bmatrix} = \begin{bmatrix} A - K\underline{b}\underline{k}^T & \vdots & \underline{b}\underline{Kc}_r^T \\ \vdots & \vdots & \vdots \\ 0 & \vdots & A_r \end{bmatrix} \begin{bmatrix} \underline{x}(t) \\ \vdots \\ \underline{x}_r(t) \end{bmatrix} \quad (34)$$

or

$$\begin{bmatrix} \dot{\underline{x}}_a(t) \end{bmatrix} = \begin{bmatrix} A_a \end{bmatrix} \begin{bmatrix} \underline{x}_a(t) \end{bmatrix} \quad (35)$$

As stated above, the solution to this augmented system is

$$\underline{x}_a(t) = e^{A_a t} \underline{x}_a(0) \quad (36)$$

The state transition matrix for the augmented system is calculated using Sylvester's expansion theorem which states that

$$e^{A_a t} = F_1 e^{\lambda_1 t} + F_2 e^{\lambda_2 t} + \dots + F_n e^{\lambda_n t} \quad (37)$$

In this equation the λ_i 's are the eigenvalues of the augmented system matrix A_a . For systems without repeated eigenvalues, the F matrices may

be found from

$$F_i = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{A_a - \lambda_j I}{\lambda_i - \lambda_j} \quad (38)$$

In this way the system states and thus the system output may be expressed as equations in terms of the eigenvalues of the original system and the applied input.

The Time Response module requires from the user the system matrices A , $\underline{b}$, $\underline{c}$, and $\underline{k}$, the initial state of the system $\underline{x}(0)$, the input transfer function polynomials $N(s)$ and $D(s)$, and the gains K and K_r . The program then forms the augmented system matrix A_a and determines its eigenvalues. If repeated eigenvalues are found, the user is provided with a listing of the eigenvalues and is returned to the local menu. If no repeated eigenvalues exist, Sylvester's expansion theorem is used to determine the F matrices relating the eigenvalues to the state transition matrix. By using Eqs. 36 and 37 the expression of each state variable is formed. Equation 26 is then used to form the expression for the output.

The Time Response module can calculate the closed-form time response for tenth order systems with a tenth order input.

4.5 Matrix Functions

The Matrix Functions module is a collection of matrix manipulation programs that calculate the determinant, inverse matrix, characteristic polynomial, eigenvalues, state transition matrix and resolvent matrices

of a given matrix. The matrix is described to the program by specifying all of its coefficients.

The determinant of the matrix is calculated by using Gaussian elimination to place the matrix in upper triangular form. Once in this form, the determinant is the product of the diagonal elements of the matrix.

The inverse matrix is calculated by using Gaussian elimination with column pivoting. Consider the original matrix A to be augmented with the identity matrix I as

$$[A : I] \quad (39)$$

By performing row operations, this augmented matrix can be placed in the form

$$[I : B] \quad (40)$$

The matrix B is then equal to the inverse of A , that is $B = A^{-1}$.

The characteristic polynomial of the matrix is calculated by the principle-minor method [6].

The eigenvalues of the matrix are calculated by first using the principle-minor method to determine the characteristic polynomial. Once the characteristic polynomial has been found, the rooting algorithms discussed under Root Locus are used to factor the polynomial and therefore determine the eigenvalues of the original matrix.

The state transition matrix is calculated by a truncated series using Eqs. 21 and 23 in the System Simulation module section.

The resolvent matrices are calculated by the Leverrier algorithm [7].

Matrices of up through twentieth order may be manipulated in the Matrix Functions module.

5.0 A PROGRAM SESSION

The purpose of this section is to provide an example session with DACS. This will aid in demonstrating the ease of use of the program and also serve to better explain some of DACS features. As an example, consider the system with the open-loop transfer function

$$G(s) = \frac{s + 20}{s^2 + 10s} \quad (41)$$

It is desired to plot the root locus of this system for gains from 0 to 75 at intervals of 1.

To begin with, the DACS program is executed on an IBM Personal Computer. Once DACS is loaded the user is greeted by the display of Fig. 6. Pressing the space bar as instructed presents the user with the main menu of Fig. 7. The user has ten options at this point, each of which is listed on the menu. If this is the user's first session, it is advantageous to turn on the HELP option by holding down the Alternate key and pressing function key one. This activates the HELP option and the menu is updated to indicate HELP is on.

The default sound level is one, which, as explained in Section 3, provides error signals only. The sound level may be changed to level two by holding down the Alternate key and pressing function key two. The menu is again updated to reflect the change. Sound level two provides both error signals and input prompt signals.

Since the problem at hand is to obtain a plot of the root locus of the system of Eq. 41, the user should press function key one to select

DACS: DESIGN AND ANALYSIS
OF LINEAR CONTROL SYSTEMS

(C) Copyright 1984
W. Ivan Lewis, III

Press Space Bar to Continue

Figure 6. DACS Opening Message

CONTROL PROGRAMS

<F1>.....ROOT LOCUS
<F2>.....FREQUENCY RESPONSE
<F3>.....SYSTEM SIMULATION
<F4>.....TIME RESPONSE
<F5>.....MATRIX FUNCTIONS
<F9>.....FILE DIRECTORY
<ESC>.....EXIT TO SYSTEM
<Alt-F1>.....HELP LEVEL OFF
<Alt-F2>.....SOUND LEVEL 1
<Alt-F3>.....ARCHIVE/ COMMENTS

PRESS DESIRED FUNCTION KEY

Figure 7. DACS Main Menu with Default Settings

the Root Locus program module. The module is loaded and the user is presented with the Root Locus menu as shown in Fig. 8.

The first order of business is for the user to describe the system of interest to DACS. By pressing function key one, the user is allowed to enter the system description into the computer. Since the HELP option is activated, DACS provides a help screen to aid in model entry. The screen appears as in Fig. 9. This figure indicates that all that is required to describe the system to DACS are the coefficients of the numerator and denominator polynomials in ascending order. By pressing a key to clear the help screen, the user is prompted for the dimensions of the numerator and denominator polynomials. In the example problem, the numerator is first order and the denominator is second order. After the user enters the required numbers, the program prompts the user for the coefficients of the denominator polynomial $D(s)$. The user is requested to enter the coefficients in ascending order pressing the carriage return after each coefficient. In the present example the only coefficient of the numerator is the constant 20. The coefficients of the denominator polynomial are 0 and 10. Since this concludes model entry, the user is again presented with the Root Locus menu of Fig. 8.

The next step is for the user to tell DACS the range of gains for which data is to be calculated. This is done under the ENTER INPUT PARAMETERS menu option, so the user should press function key five. Again, since HELP is activated, the user is presented with information to aid in parameter input. This information appears in Fig. 10. The program needs to know the range of gains of interest and the gain step

ROOT LOCUS

<F1>.....ENTER NEW MODEL
<F2>.....RECALL OLD MODEL
<F3>.....MODIFY CURRENT MODEL
<F4>.....STORE MODEL ON DISK
<F5>.....ENTER INPUT PARAMETERS
<F6>.....CALCULATE DATA
<F7>.....GRAPH DATA
<F8>.....LIST/SAVE DATA
<F9>.....FILE DIRECTORY
<ESC>.....EXIT TO MAIN MENU
<Alt-F1>.....HELP LEVEL ON
<Alt-F2>.....SOUND LEVEL 2
<Alt-F3>.....ARCHIVE/COMMENTS

PRESS DESIRED FUNCTION KEY

Figure 8. Root Locus Menu

The Root Locus program calculates the zeros of the equation

$$1 + G(s)$$

as a function of K. $G(s)$ is assumed rational and

$$G(s) = KN(s)/D(s)$$

You will be prompted for the polynomials $N(s)$ & $D(s)$.
The coefficient of the highest order term is assumed unity.
Enter the coefficients in ascending order. Example:

$$G(s) = (s + 3)/(s^3 + 2s^2 + 9s + 3)$$

$$N(s) = 3$$

$$D(s) = 3 \quad 9 \quad 2$$

The coefficient of the highest order term is not entered.

Press any key.....

Figure 9. Help Screen on Root Locus Model Entry

You will be asked to enter the minimum gain, KMIN, the maximum gain, KMAX, and the gain interval, KINT. The number of points to be calculated equals

$$(KMAX - KMIN) / KINT.$$

This number may not exceed 500

Press any key.....

Figure 10. Help Screen on Root Locus Parameter Input

size to use in the calculations. By pressing a key, the user is prompted for the minimum gain, maximum gain, and gain interval. The current problem is to plot the root locus for gains from 0 to 75 at a gain interval of 1. This concludes parameter input and the user is returned to the Root Locus menu.

The system is now completely described by the user and data is ready to be calculated. By pressing function key six, the program begins to calculate the requested data and the screen displays the message "CALCULATING DATA.....". Since the current system is second order, all roots will be calculated in closed-form and all 150 roots will be calculated in approximately 3 seconds. When calculations are complete, the user will again be returned to the Root Locus menu.

To obtain a plot of the locus, the user selects the GRAPH DATA option by pressing function key seven. The user is presented with the display screen of Fig. 11. The user is interested in plotting the data just calculated which is accomplished by holding down the control key and pressing function key one. The user is asked if grids are desired on the graph. A "no" answer results in the curve of Fig. 12 which is the desired root locus plot.

Once the graph is displayed, the user has several options. One of these is to obtain coordinate information about any point on the plot. By using the cursor keys, the cursor at the origin may be moved to any point within the boundaries of the graph. Once at the desired point, pressing the carriage return causes the graph coordinates corresponding to the cursor position to be displayed in the upper right hand corner of

OPTIONS

<Ctrl-F1>.....GRAPH CURRENT DATA
<Ctrl-F2>.....RE-DISPLAY SCREEN FILE
<ESC>.....END

WHILE GRAPH DISPLAYED

<ENTER>.....CURSOR COORDINATES
<Ctrl-F4>.....SAVE GRAPH ON DISK
<Ctrl-F5>.....ZOOM
<ESC>.....END

(EXCEPT ON RE-DISPLAY)

Figure 11. Root Locus Graphing Options

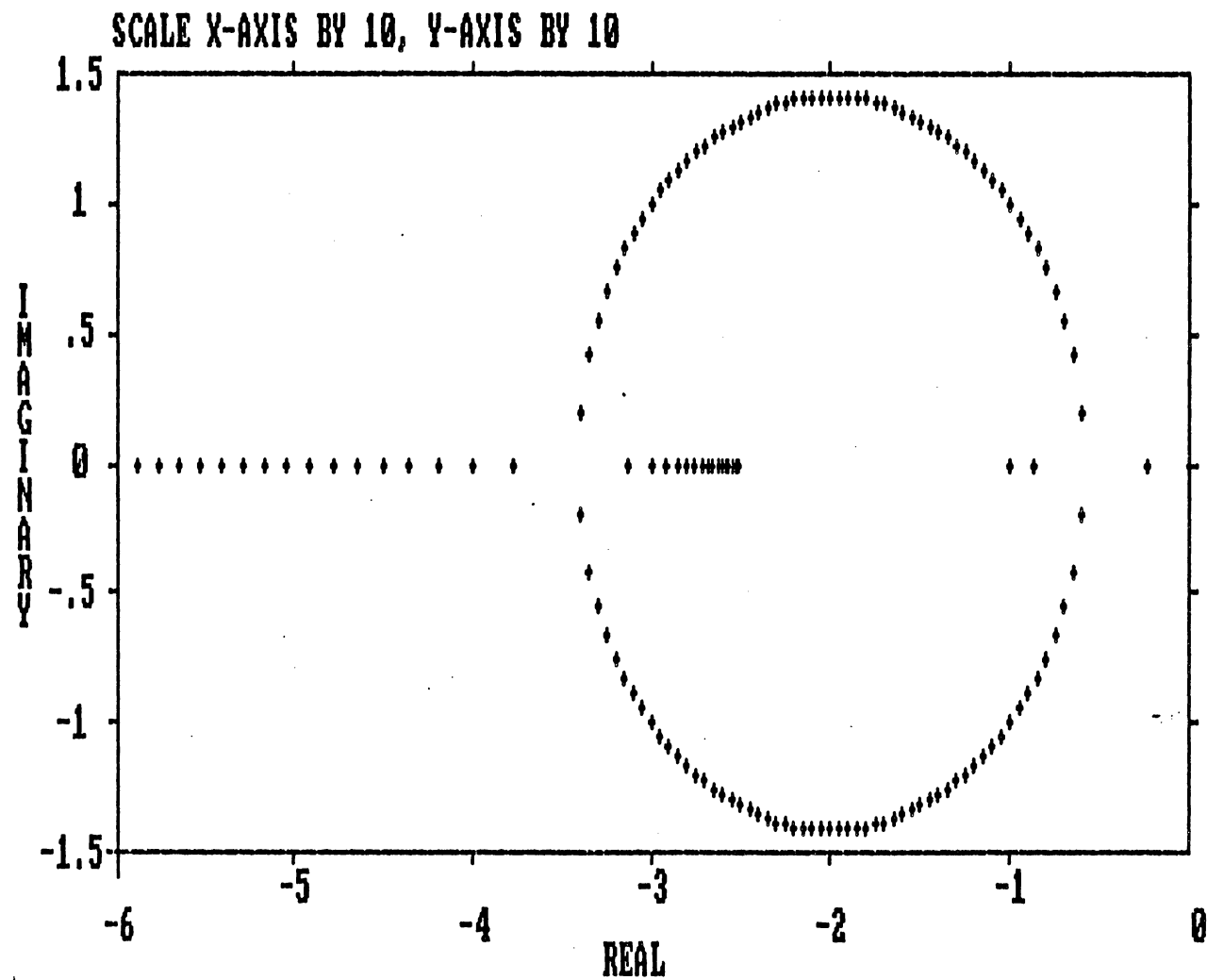


Figure 12. Example System Root Locus

the screen. An example is shown in Fig. 13. Other options available to the user at this time are printing the graph, zooming in or out on the graph, and saving the graph on disk so that it can be redisplayed at a later time. When the user is finished analyzing the plot, pressing the escape key will return the user to the graphing menu. Pressing the escape key again will return the user to the Root Locus menu.

The user may wish to view the calculated data directly versus viewing it through a plot. This can be accomplished by selecting the LIST/SAVE DATA option with function key eight. The user is presented with the data menu of Fig. 14. By selecting LIST CURRENT DATA with the key combination control - function key six, the screen display of Fig. 15 appears. This display shows the two roots of the system calculated for a gain of zero. By pressing a key the user is presented with the next two roots calculated for a gain of 1. The user may continue in this manner, displaying the roots for each gain, or the escape key may be pressed to return the user to the data menu. Pressing the escape key again returns the user to the Root Locus menu.

Before leaving the Root Locus module, the user may want to save the model in a file on disk so that it can be re-analyzed later. This option is available under the menu selection SAVE MODEL. By pressing function key four, the user is presented with file information and asked if a file directory is desired, Fig. 16. Answering "no", the user is prompted for the name of the file in which the model is to be saved. During the saving process the message "SAVING MODEL....." is displayed. After the model is saved, the user is returned to the Root

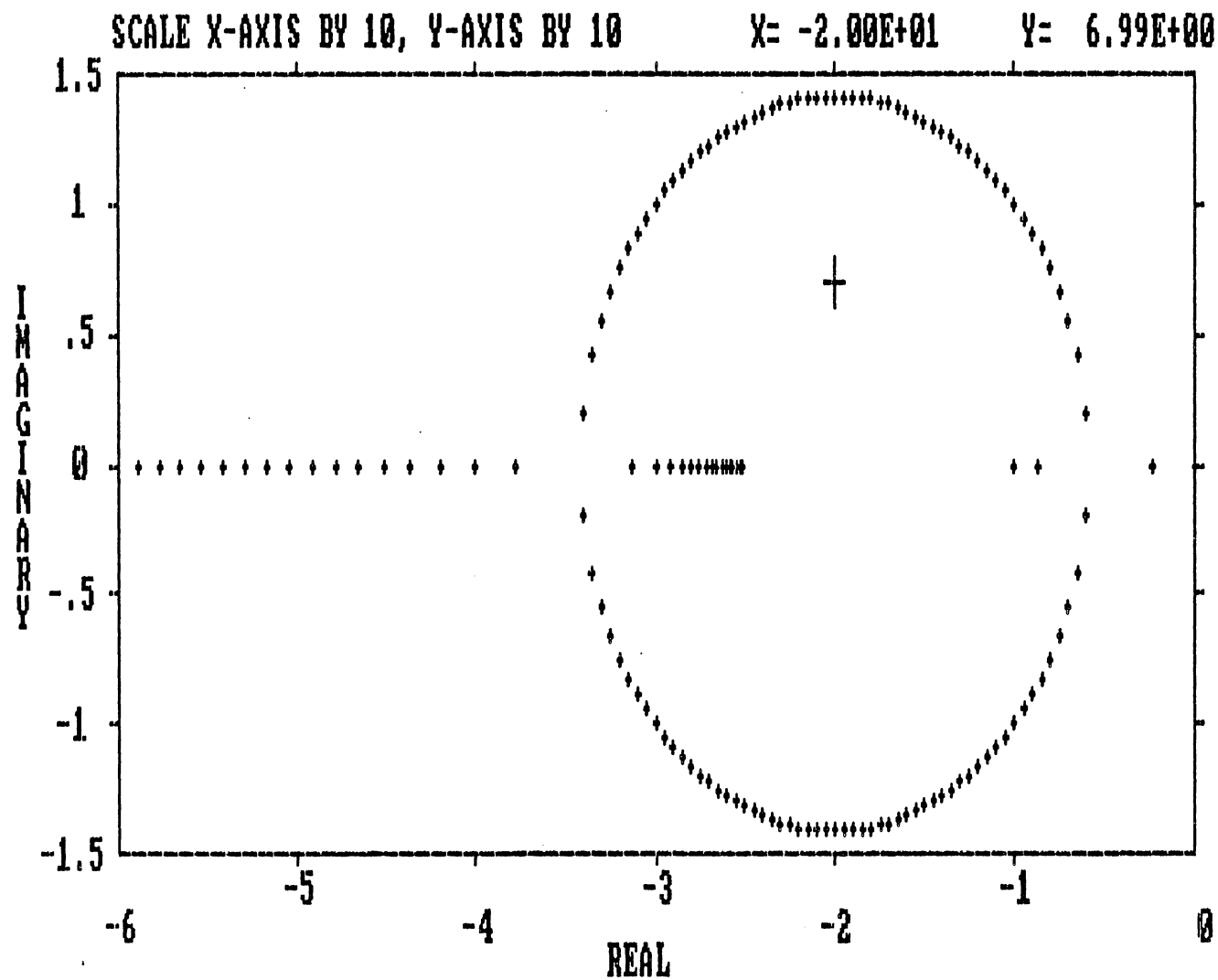


Figure 13. Example System Root Locus Showing Cursor Option

OPTIONS

<Ctrl-F6>.....LIST CURRENT DATA
<Ctrl-F7>.....SAVE DATA ON DISK
<Ctrl-F8>.....RECALL DATA FILE
<ESC>.....END

Figure 14. Root Locus Data Menu

GAIN = 0

REAL	IMAGINARY
0	0
-10	0

Press any key.....

Figure 15. Example System Roots for $K = 0$

A filename is 1-8 letters/digits. A filename may
be preceded by a disk drive letter and a colon,
otherwise the current logged drive is used.

WARNING! Using an existing filename will erase the existing file.

Would you like to see your
File Directory? (Y/N)

Figure 16. Root Locus File Help Screen

Locus menu.

The user may stay within the Root Locus module as long as desired and continue to analyze the present system or enter new systems to be analyzed. Whenever the user is finished performing root locus calculations, the escape key provides an exit to the main menu.

All program modules are similar in appearance and operation to the Root Locus module. Figures 17 through 20 show the local menus of the Frequency Response, System Simulation, Time Response, and Matrix Functions modules.

FREQUENCY RESPONSE

<F1>.....ENTER NEW MODEL
<F2>.....RECALL OLD MODEL
<F3>.....MODIFY CURRENT MODEL
<F4>.....STORE MODEL ON DISK
<F5>.....ENTER INPUT PARAMETERS
<F6>.....CALCULATE DATA
<F7>.....GRAPH DATA
<F8>.....LIST/ SAVE DATA
<F9>.....FILE DIRECTORY
<ESC>.....EXIT TO MAIN MENU
<Alt-F1>.....HELP LEVEL OFF
<Alt-F2>.....SOUND LEVEL 1
<Alt-F3>.....ARCHIVE/COMMENTS

PRESS DESIRED FUNCTION KEY

Figure 17. Frequency Response Menu

SYSTEM SIMULATION

<F1>.....ENTER NEW MODEL
<F2>.....RECALL OLD MODEL
<F3>.....MODIFY CURRENT MODEL
<F4>.....STORE MODEL ON DISK
<F5>.....ENTER INPUT PARAMETERS
<F6>.....CALCULATE DATA
<F7>.....GRAPH DATA
<F8>.....LIST/ SAVE DATA
<F9>.....FILE DIRECTORY
<ESC>.....EXIT TO MAIN MENU
<Alt-F1>.....HELP LEVEL OFF
<Alt-F2>.....SOUND LEVEL 1
<Alt-F3>.....ARCHIVE/COMMENTS

PRESS DESIRED FUNCTION KEY

Figure 18. System Simulation Menu

TIME RESPONSE

<F1>.....ENTER NEW MODEL
<F2>.....RECALL MODEL/INPUT
<F3>.....MODIFY MODEL/INPUT
<F4>.....STORE MODEL/INPUT
<F5>.....ENTER INPUT PARAMETERS
<F6>.....CALCULATE DATA
<F8>.....LIST/SAVE/RECALL DATA
<F9>.....FILE DIRECTORY
<ESC>.....EXIT TO MAIN MENU
<Alt-F1>.....HELP LEVEL OFF
<Alt-F2>.....SOUND LEVEL 1
<Alt-F3>.....ARCHIVE/ COMMENTS

PRESS DESIRED FUNCTION KEY

Figure 19. Time Response Menu

MATRIX FUNCTIONS

<F1>.....ENTER NEW MATRIX
<F2>.....RECALL OLD MATRIX
<F3>.....MODIFY CURRENT MATRIX
<F4>.....STORE MATRIX ON DISK
<F6>.....CALCULATE DATA
<F9>.....FILE DIRECTORY
<ESC>.....EXIT TO MAIN MENU
<Alt-F1>.....HELP LEVEL OFF
<Alt-F2>.....SOUND LEVEL 1
<Alt-F3>.....ARCHIVE/ COMMENTS

PRESS DESIRED FUNCTION KEY

Figure 20. Matrix Functions Menu

6.0 DESIGN EXAMPLES

This section is designed to show how DACS can be used to analyze control systems.

6.1 Servo Positioning System

As an example of the use of DACS in control system design, consider the typical servo system shown in Fig. 21. The figure depicts a positioning system in which the actual position is compared with the desired position to generate an error signal. This error signal is fed into the controller to help correct a positioning discrepancy. The system has the block diagram of Fig. 22. Assuming for the moment that there is no disturbance input to the system, block diagram techniques may be used to arrive at the reduced system of Fig. 23 which has the open-loop transfer function

$$G(s) = \frac{8K(s + 5)}{s^4 + 13s^3 + 48s^2 + 180s} \quad (42)$$

Using the block diagram of Fig. 22 we may also determine a state model for the system as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 4 & 0 & 0 \\ 0 & -2 & 2 & 0 \\ -K & -8 & -1 & -5 \\ -K & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -1 \\ K & 0 \\ K & 0 \end{bmatrix} \begin{bmatrix} u_2 \\ u_2 \end{bmatrix} \quad (43)$$

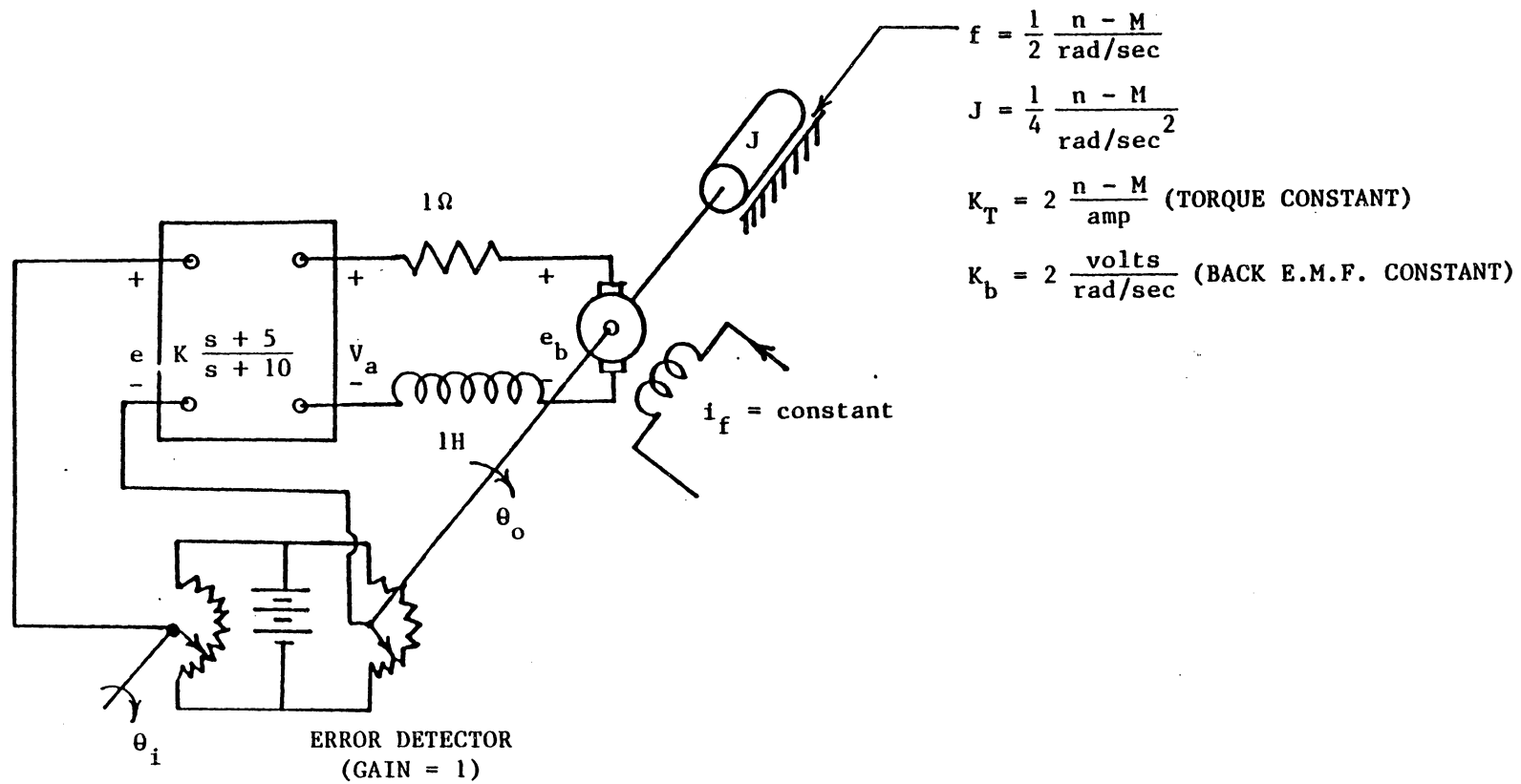


Figure 21. Servo System

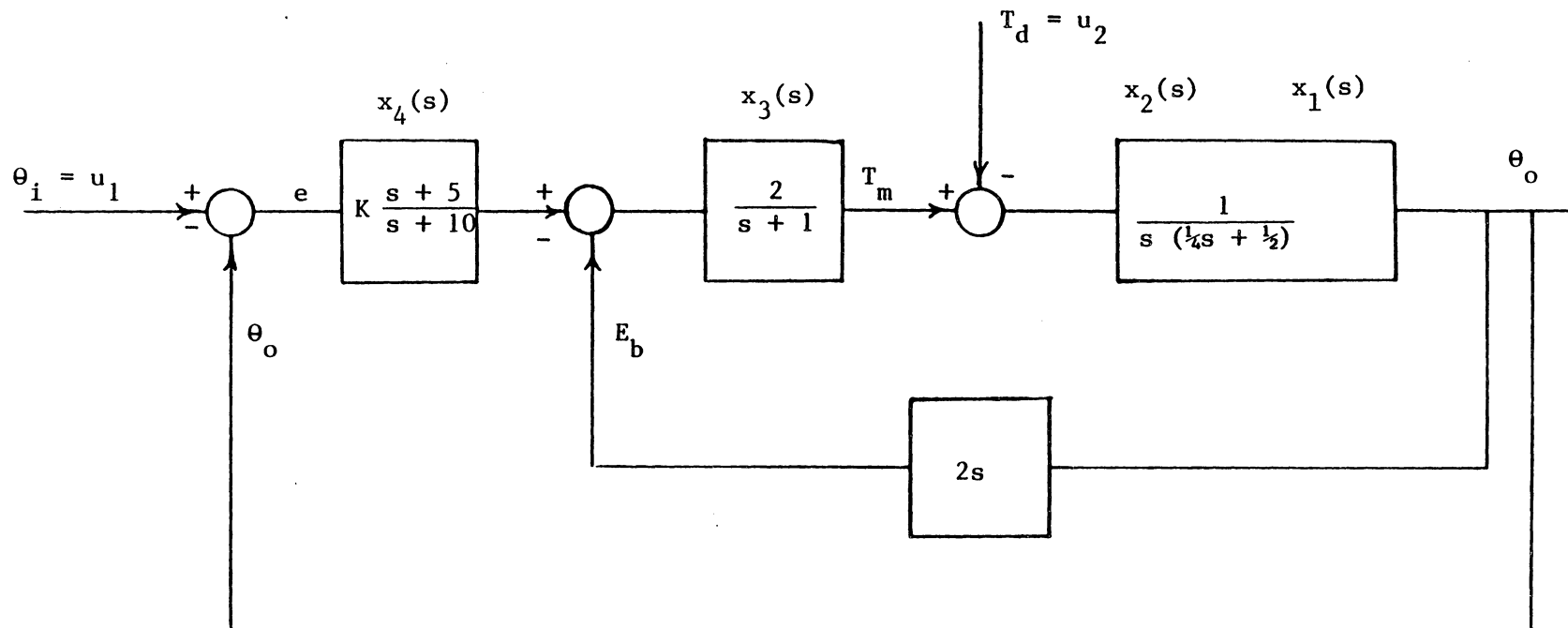


Figure 22. Servo System Block Diagram

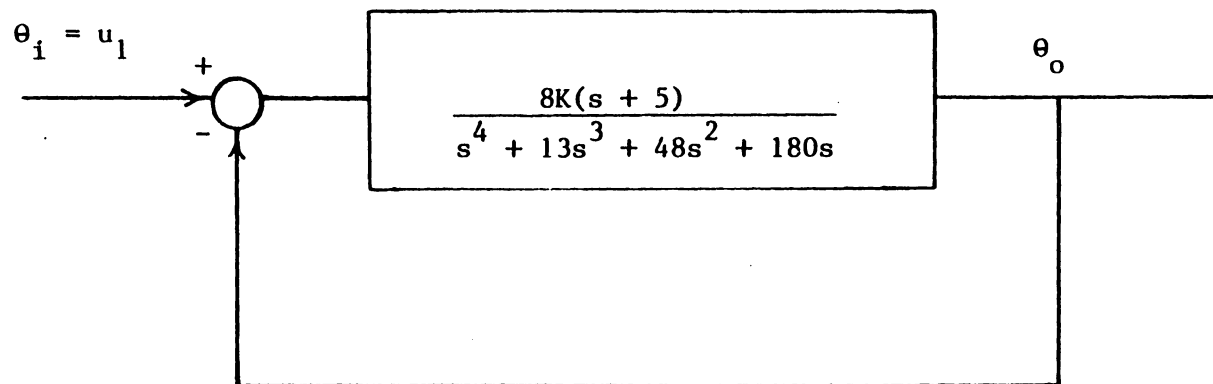


Figure 23. Reduced Servo Diagram

where the output becomes

$$y(t) = \theta_0(t) = x_1(t) \quad (44)$$

Now that both a system transfer function and a system state model are available, DACS may be used to investigate the influence of gain on the system. First the Root Locus module is used to try and establish the range of stable gains for the system. Entering the open-loop transfer function and calculating the locus for gains from 1 to 20 results in Fig. 24. By examining this figure and the calculated data, it can be determined that the system is stable for gains of 1 to approximately 14.

The simulation module can be used next to graphically display the time response of the system for gains of $K = 1, 4, 5, 10$, and 20. Using a unit step input and no disturbance, the plots of Figs. 25 through 29 are generated. From these it can be seen that increased gain results in quicker system response but with increased ringing. For the gain of 20 it is clear that the system is unstable as predicted. Choosing the response with $K = 4$ to meet the system specifications, the influence of disturbance inputs can be investigated by allowing u_2 to be nonzero. Figure 30 was generated for $K = 4$, $u_1 = 1$ and $u_2 = 5 \sin(377t)$ (60 cycle noise). The disturbance input has no apparent effect on the system.

Finally, by using the Time Response module, the closed-form time response for each state of the system and the system output can be found. For a unit step input with $K = 4$, it was found that

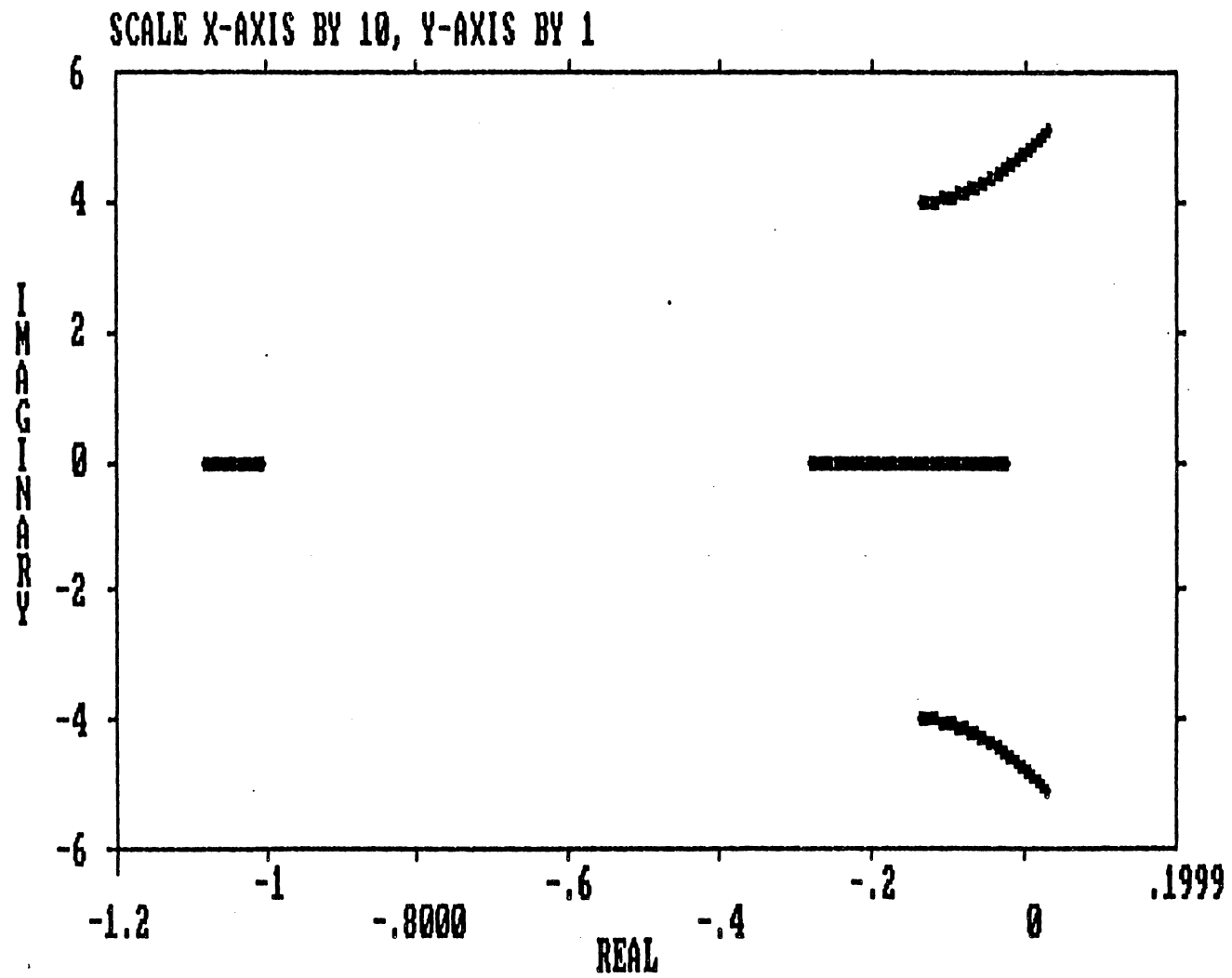


Figure 24. Servo Root Locus ($K = 1 - 20$)

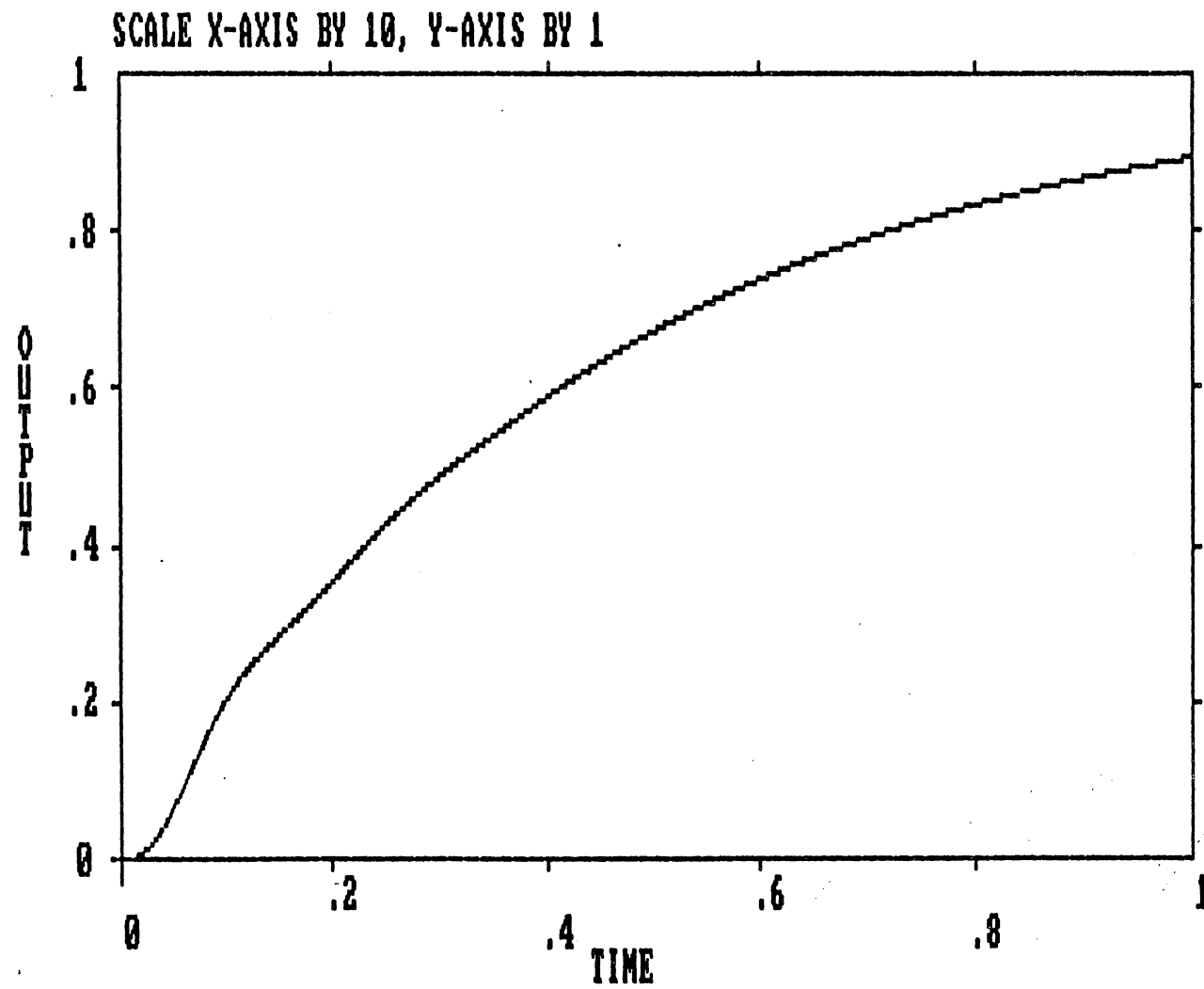


Figure 25. Servo Step Response ($K = 1$)

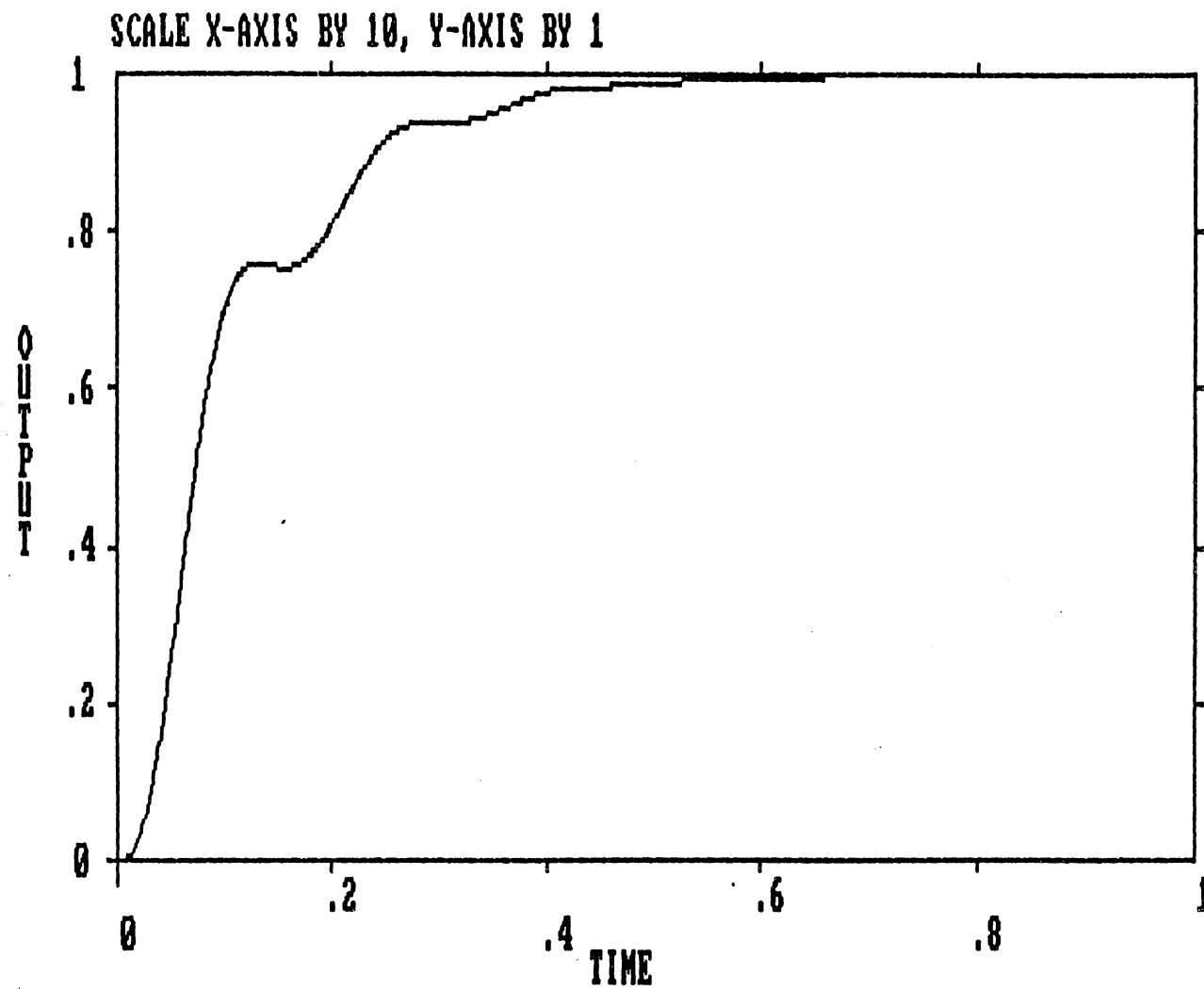


Figure 26. Servo Step Response ($K = 4$)

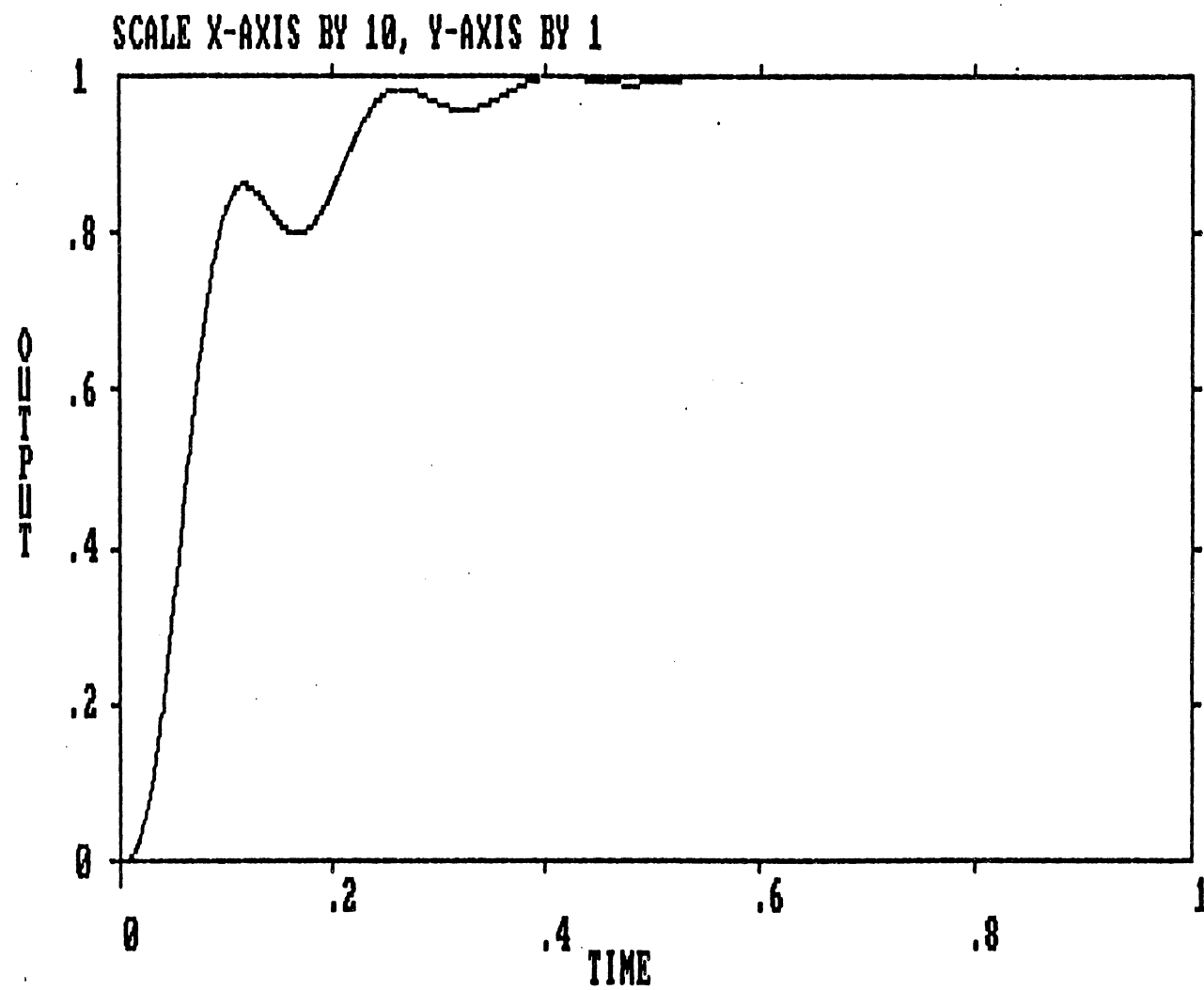


Figure 27. Servo Step Response ($K = 5$)

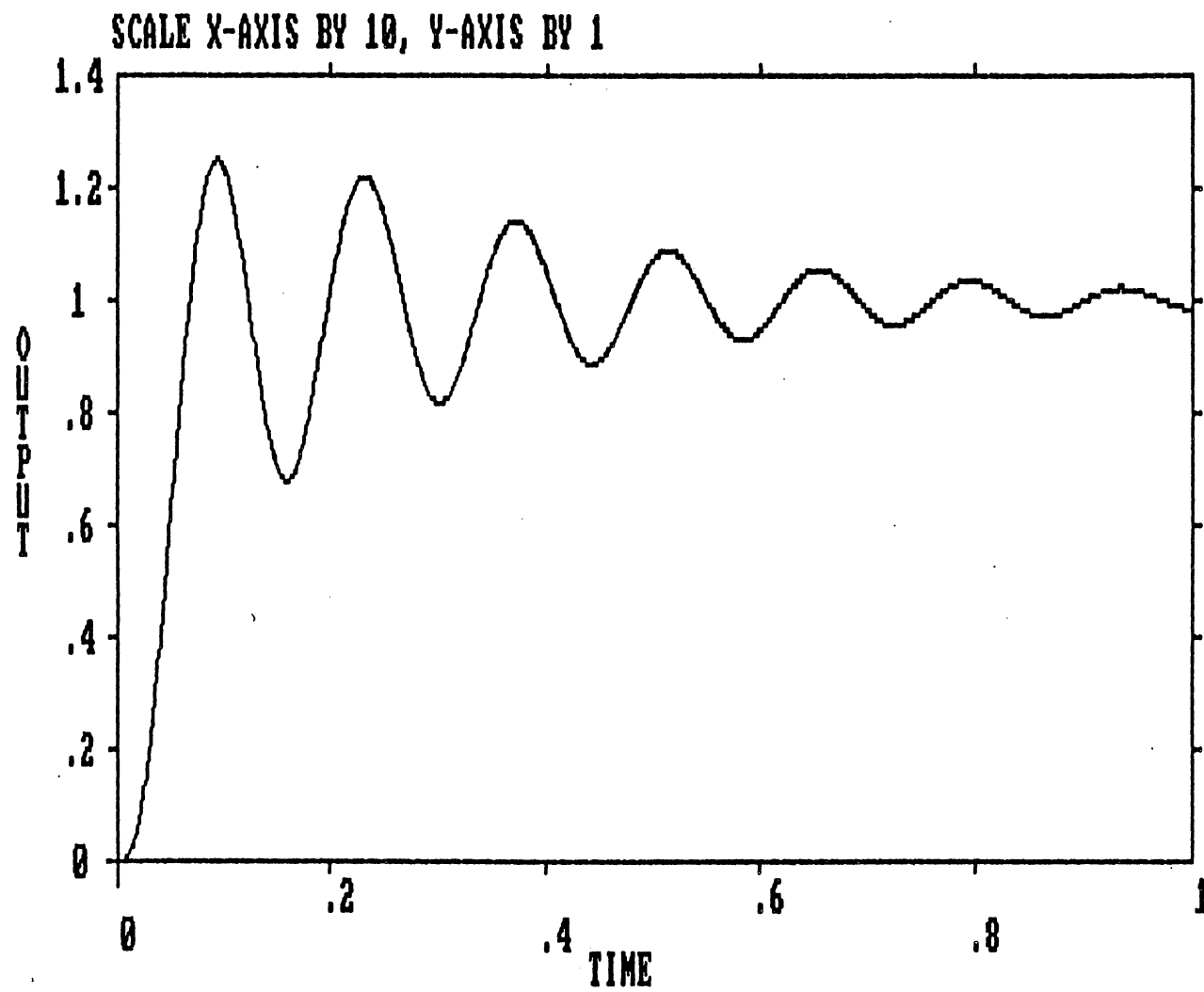


Figure 28. Servo Step Response ($K = 10$)

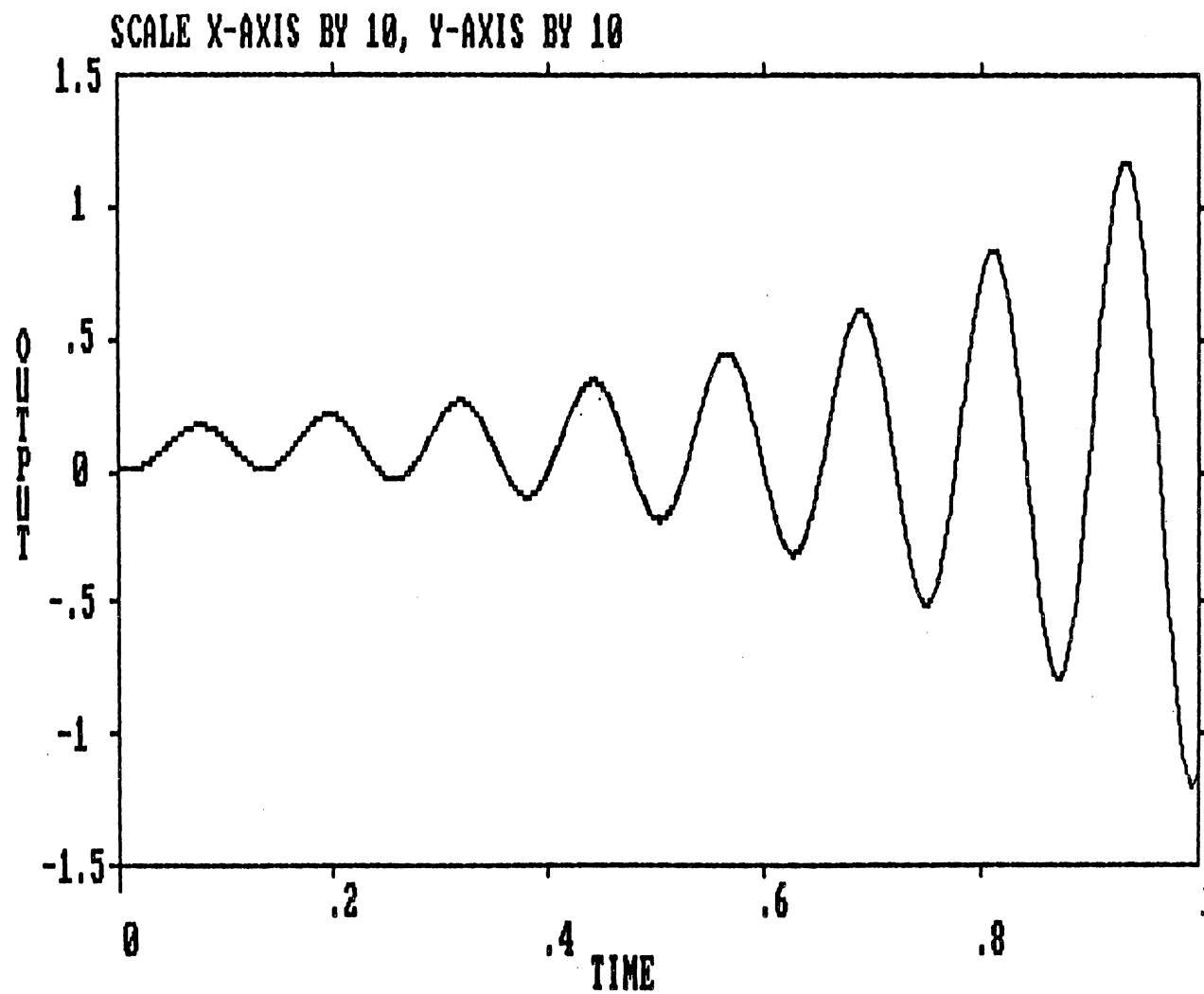


Figure 29. Servo Step Response ($K = 20$)

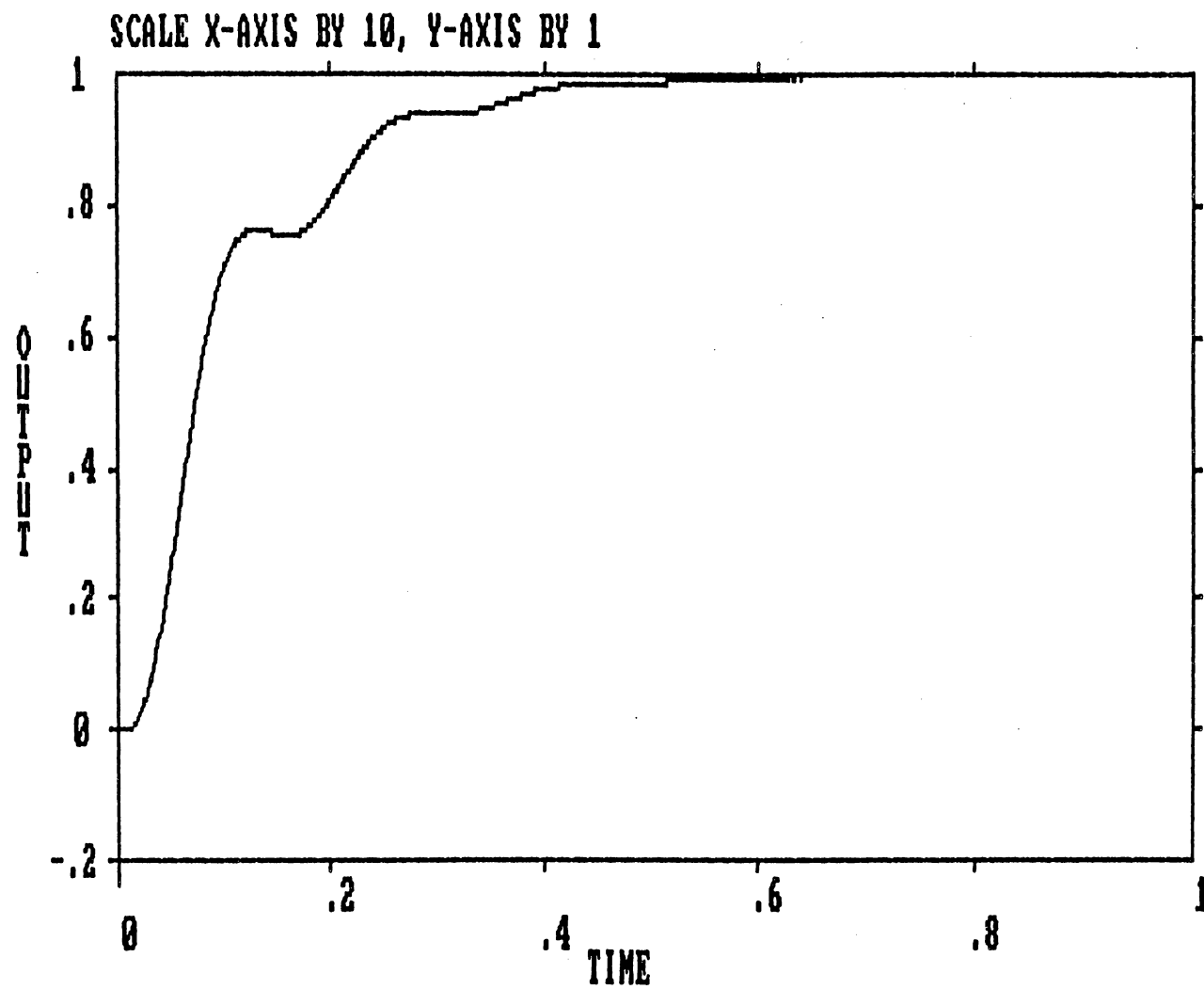


Figure 30. Servo Step Response with Disturbance

$$\begin{aligned}
x_1(t) &= - .016e^{-.963t} \cos(4.078t) - .033e^{-.963t} \sin(4.078t) \\
&\quad + .259e^{-10.179t} - .950e^{-.895t} + 1 \\
x_2(t) &= - .128e^{-.963t} \cos(4.078t) - .257e^{-.963t} \sin(4.078t) \\
&\quad - .096e^{-10.179t} + .213e^{-.895t} \\
x_3(t) &= .031e^{-.963t} \cos(4.078t) + .062e^{-.963t} \sin(4.078t) \\
&\quad - .573e^{-10.179t} + .177e^{-.895t} \\
x_4(t) &= -.015e^{-.963t} \cos(4.078t) - .031e^{-.963t} \sin(4.078t) \\
&\quad -.101e^{-10.179t} + .417e^{-.895t}
\end{aligned} \tag{45}$$

and

$$y(t) = x_1(t) \tag{46}$$

6.2 Electro-Hydraulic Ship Steering System

As a second example of the use of DACS to investigate linear control systems, consider the electro-hydraulic ship steering system of Fig. 31. The wheel position is sensed by a linear variable differential transformer (LVDT) which supplies an electrical signal to the amplifier, which in turn drives the electro-hydraulic servo valve. The valve controls the stroke of the variable displacement pump which permits both positive and negative stroke positions, meaning the output flow can be

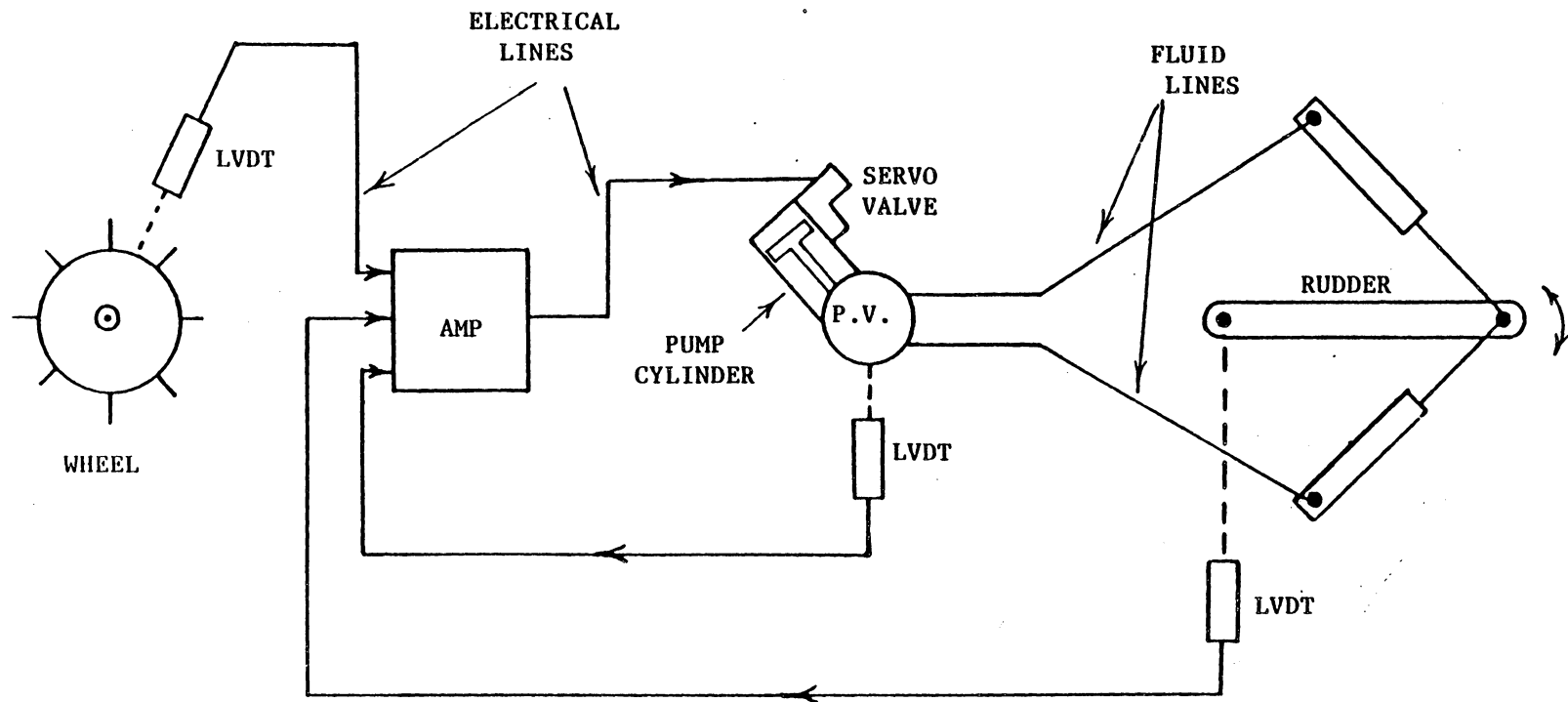


Figure 31. Electro-Hydraulic Ship Steering System

reversed in direction. Fluid from the pump enters one of the two rudder cylinders to control the rudder position. A second LVDT feeds back rudder position while a third LVDT supplies pump-stroke information to the amplifier. The block diagram for the system is given in Fig. 32. Using Mason's formula, the system transfer function can be determined as

$$G(s) = \frac{K_1 K_2 K_3 K_4}{\tau s^3 + s^2 + K_1 K_2 K_5 s + K_1 K_2 K_3 K_4 K_6} \quad (47)$$

Consider the system where $\tau = 1$, $K_1 = K_2 = K_3 = K_4 = 1$, $K_5 = K_6 = -2$. The system transfer function then becomes

$$G(s) = \frac{1}{s^3 + s^2 - 2s - 2} \quad (48)$$

and the system differential equation is

$$\ddot{c}(t) + \dot{c}(t) - 2\dot{c}(t) - 2c(t) = r(t) \quad (49)$$

Choosing the three system states to be rudder position, rudder velocity, and rudder acceleration, the input to be wheel position and the output to be rudder position results in the following state model:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r(t) \quad (50)$$

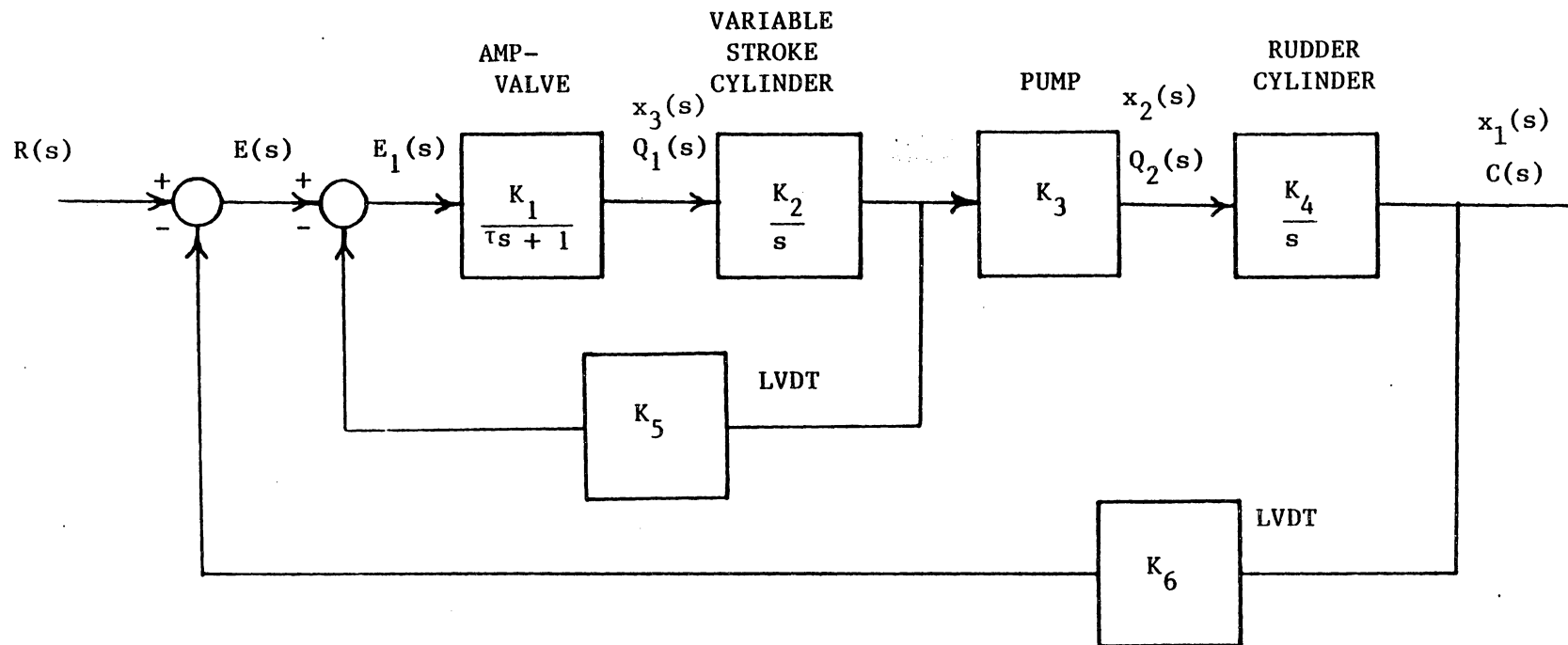


Figure 32. Steering System Block Diagram

There are several ways in which the stability of this system could be investigated using DACS. Using the Matrix Functions module, the eigenvalues of the system matrix can be found to be

$$\begin{aligned}\lambda_1 &= 1.414214 \\ \lambda_2 &= -1.414214 \\ \lambda_3 &= -1\end{aligned}\tag{53}$$

Thus there is an eigenvalue in the right-half plane and the system is unstable. If the rudder of this ill-fated vessel has an initial position of 1 and an initial velocity of 1 but no input at the wheel ($r(t)=0$), the system simulation module can be used to find the time response of the output $y(t) = \underline{c}^T \underline{x}(t)$. From Fig. 33 it is seen that the output $y(t)$ (rudder position) is unstable and increases exponentially making the ship turn faster and faster to one side. Eventually the rudder will reach its limit of travel and the vessel will sail in a circle. This indicates that there is a problem with the original choice of the system gains and DACS should be used to design a more appropriate system.

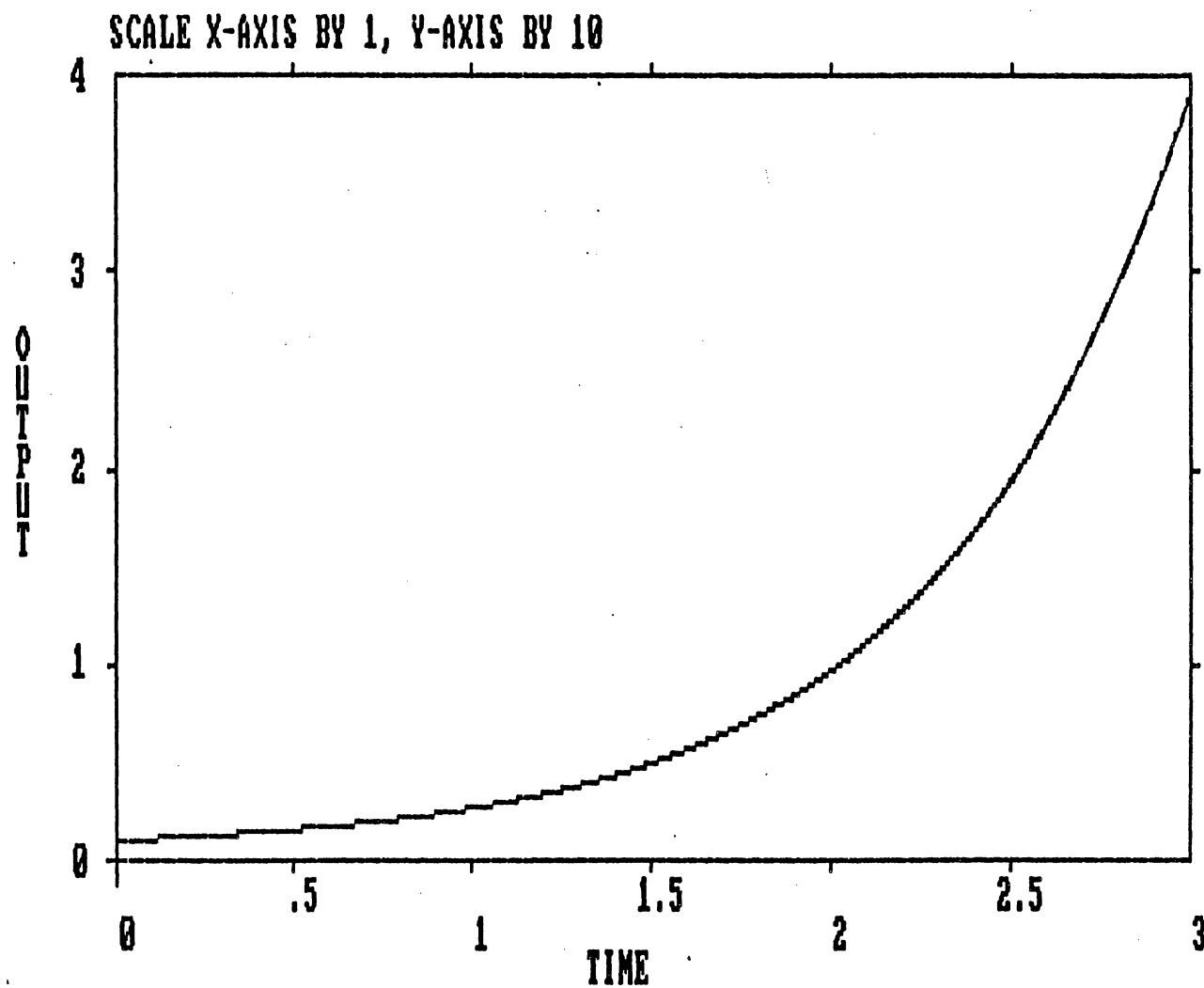


Figure 33. Steering System Response

7.0 CONCLUSIONS AND RECOMMENDATIONS

With the recent rise in computer hardware technology, more and more engineers are turning to desktop computers as a means of expediting engineering problem solving. This increase in usage has, in turn, created a need for engineering software for these desktop computers.

This thesis has presented an interactive computer program designed to address the above need for linear control systems. The program, DACS (Design and Analysis of Control Systems), was developed for the IBM Personal Computer. Besides being extremely easy to use, DACS provides a variety of solution formats. Solutions can be represented graphically, numerically, or analytically. DACS is a useful research, teaching, and design tool for linear control systems.

In spite of its power and ease of use, there is always room for improvement. One area that could be strengthened to enhance DACS is in the area of printer support. DACS already allows the user to output a graph to the printer, a very helpful capability. At the present, however, a printer interface for data output is not included. Adding an option to each local menu that, when selected, would drive a printer to tabularly list calculated data would add to DACS usefulness.

A second area where DACS could be improved is in its iterative rooting routine. As explained in section 4.1, DACS uses a Bairstow rooting algorithm to find the roots of a polynomial of degree five or greater. The routine is often slow and, for ill-behaved systems, may blow up. Extra statements could be added to the routine to speed convergence and to trap nonconvergence errors.

Other features such as symbolic input and allowing transfer functions to be input in factored form would also serve to make DACS more powerful and easier to use.

REFERENCES

1. "Continuous System Modeling Program III (CSMP III) Program Reference Manual," IBM Corporation, #SH19-7001, 1972.
2. "Advanced Continuous Simulation Language (ACSL) User Guide/Reference Manual," Mitchell and Gauthier, Associates, Inc., 290 Baker Avenue, Concord, Mass., 1981.
3. "Super-Sceptre, A Program for the Analysis of Electrical, Mechanical, Digital, and Control Systems," NTIS #AD-A011 348, May 1975.
4. "Dynamic Simulation Language/VS Language Reference Manual," IBM Corporation, #SH20-6288, June 1984.
5. Osgood, D., "A Computer on Every Desk - A Survey of Personal Computers in Universities," BYTE-The Small Systems Journal, Vol. 9, No. 6, June 1984.
6. "An Algorithm for the Analysis and Design of Linear State Variable Feedback Systems," J. L. Melsa, Proc. of Asilomar Conf. on Systems and Circuits, Monterey, California, Nov. 1967, pp. 791-799.
7. Zadeh, L. A., and C. A. Desoer, Linear System Theory, McGraw-Hill Book Co., Inc., 1963, pp. 303-305.

APPENDIX A: DACS CALCULATION EXAMPLES

This section contains examples calculated using the different modules of DACS.

A.1 Root Locus

Consider the following open-loop transfer function

$$G(s) = \frac{s + 20}{s^2 + 10s} \quad (52)$$

The root locus is to be plotted for gains from 0 to 75 in steps of 1.

The required user input is summarized as

N(s) DIMENSION >	1	
D(s) DIMENSION >	2	
N(s) POLYNOMIAL >	20	
D(s) POLYNOMIAL >	0	10
MINIMUM GAIN >	0	
MAXIMUM GAIN >	75	
GAIN INTERVAL >	1	

Entering these parameters in response to the DACS prompts and plotting the calculated data results in the screen display of Fig. 34 which is the desired root locus.

A.2 Frequency Response

Consider the open-loop transfer function

$$G(s) = \frac{s}{s + 1} \quad (53)$$

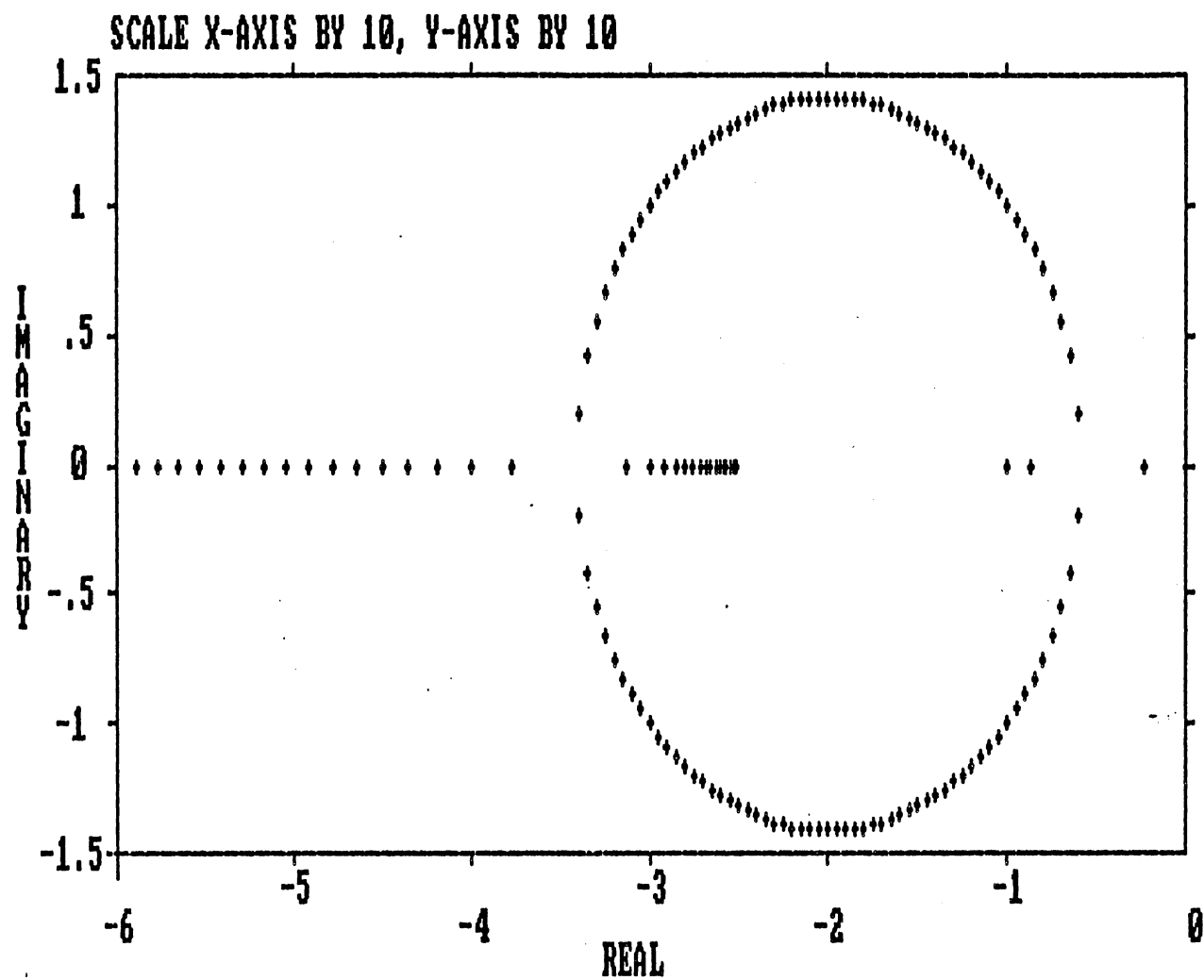


Figure 34. Root Locus for Equation 52

It is desired to obtain a Bode plot, Nyquist diagram, and Nichols chart for this transfer function over the frequency range 0.1 to 100 radians/second. The following information is entered in response to DACS prompts

```

N(s) DIMENSION >          1
D(s) DIMENSION >          1
N(s) POLYNOMIAL >         0
D(s) POLYNOMIAL >         1
SYSTEM GAIN >              1
MINIMUM FREQ >             .1
MAXIMUM FREQ >            100
NUMBER FREQS >            1000

```

By performing Bode, Nyquist and Nichols calculations and plotting the results, the screen displays of Figs. 35 through 39 are obtained. Figure 35 is the Bode plot with magnitude and phase displayed on the same screen. Figures 36 and 37 are the magnitude and phase displayed separately. Figure 38 is the generated Nyquist diagram and Fig. 39 is the generated Nichols chart.

A.3 System Simulation

Consider a system which has a state model of the form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1600 \\ 1 & 0 & -100 \\ 0 & 1 & -20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1600 \\ 0 \\ 0 \end{bmatrix} u \quad (54)$$

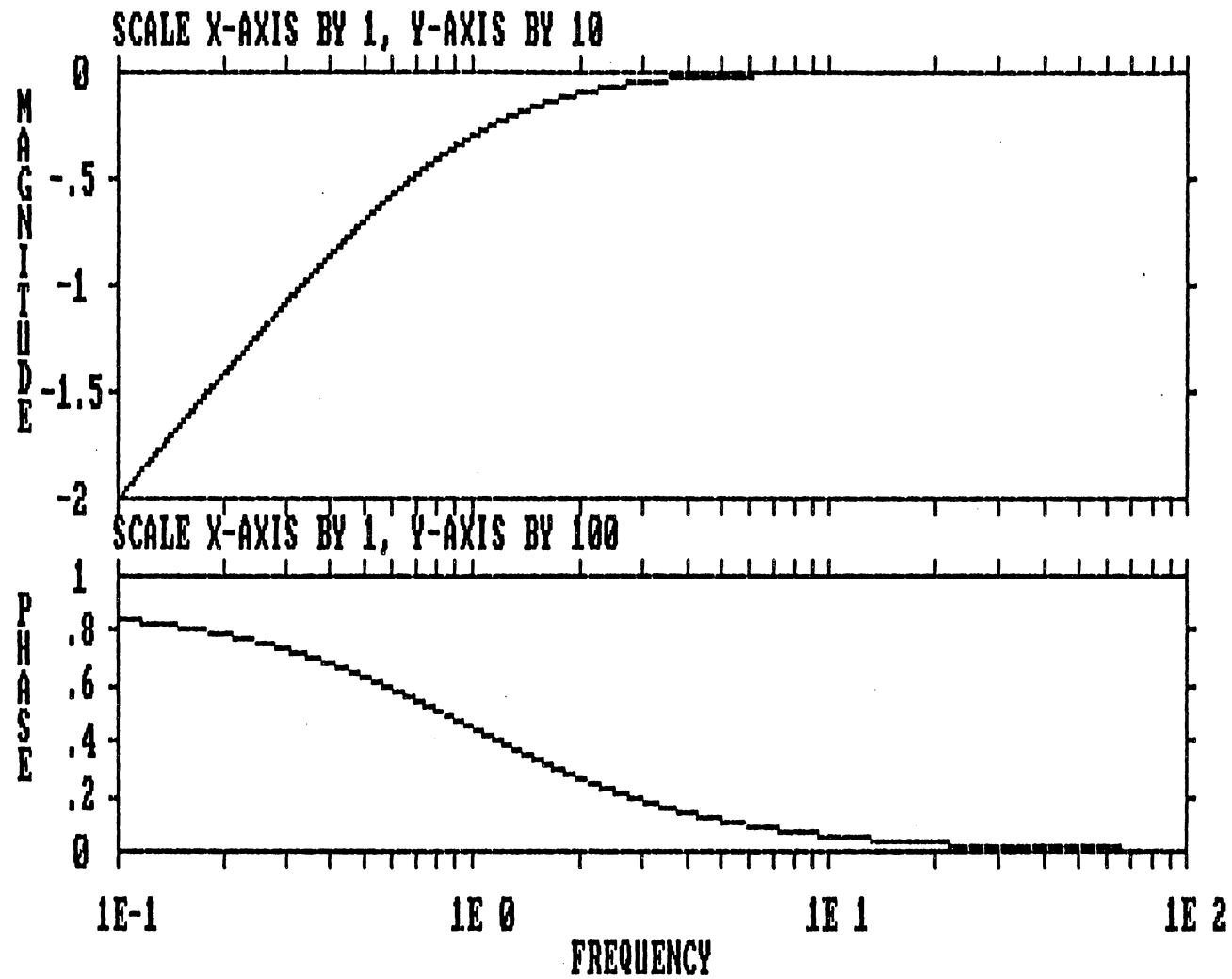


Figure 35. Bode Amplitude and Phase Example

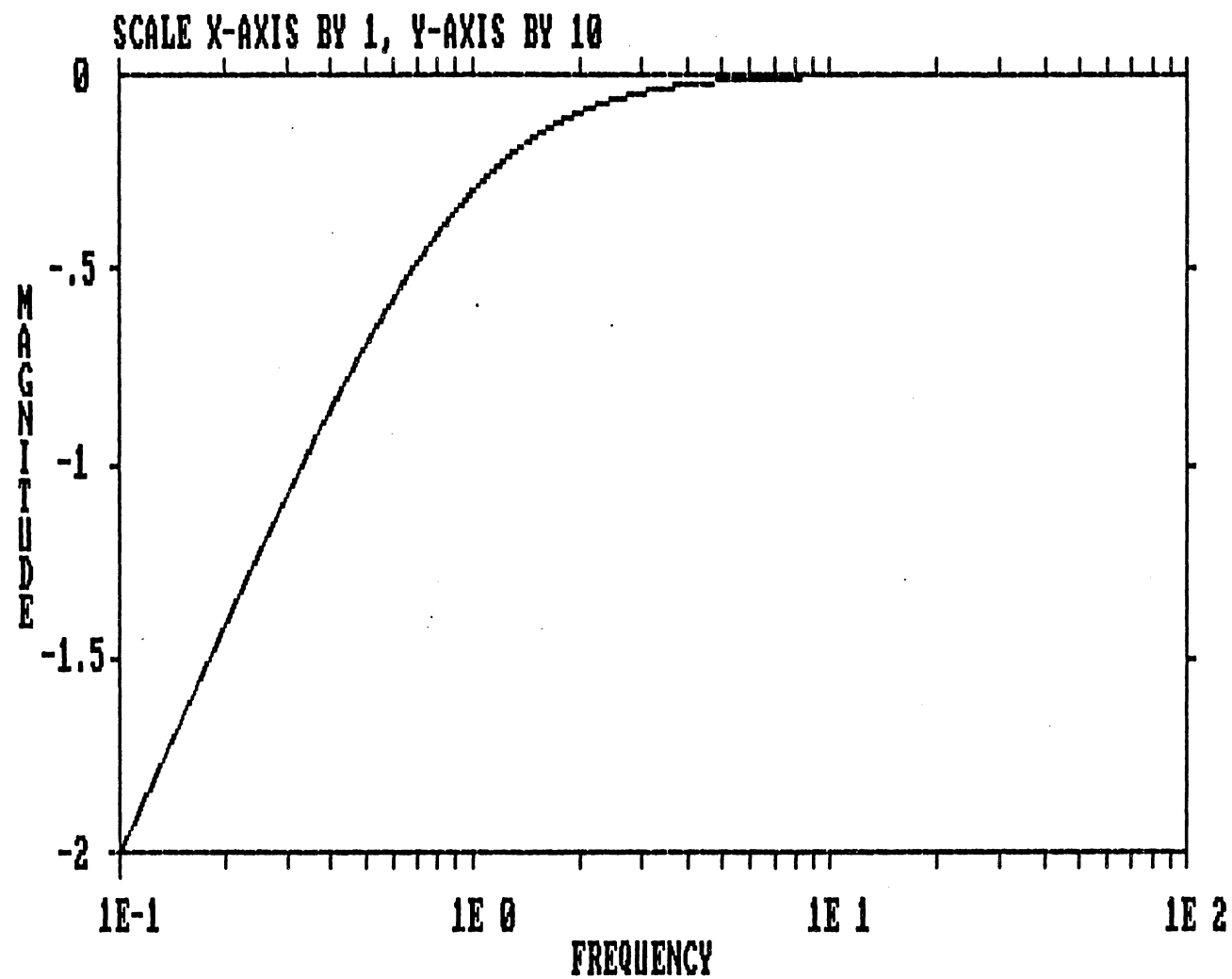


Figure 36. Bode Magnitude Example

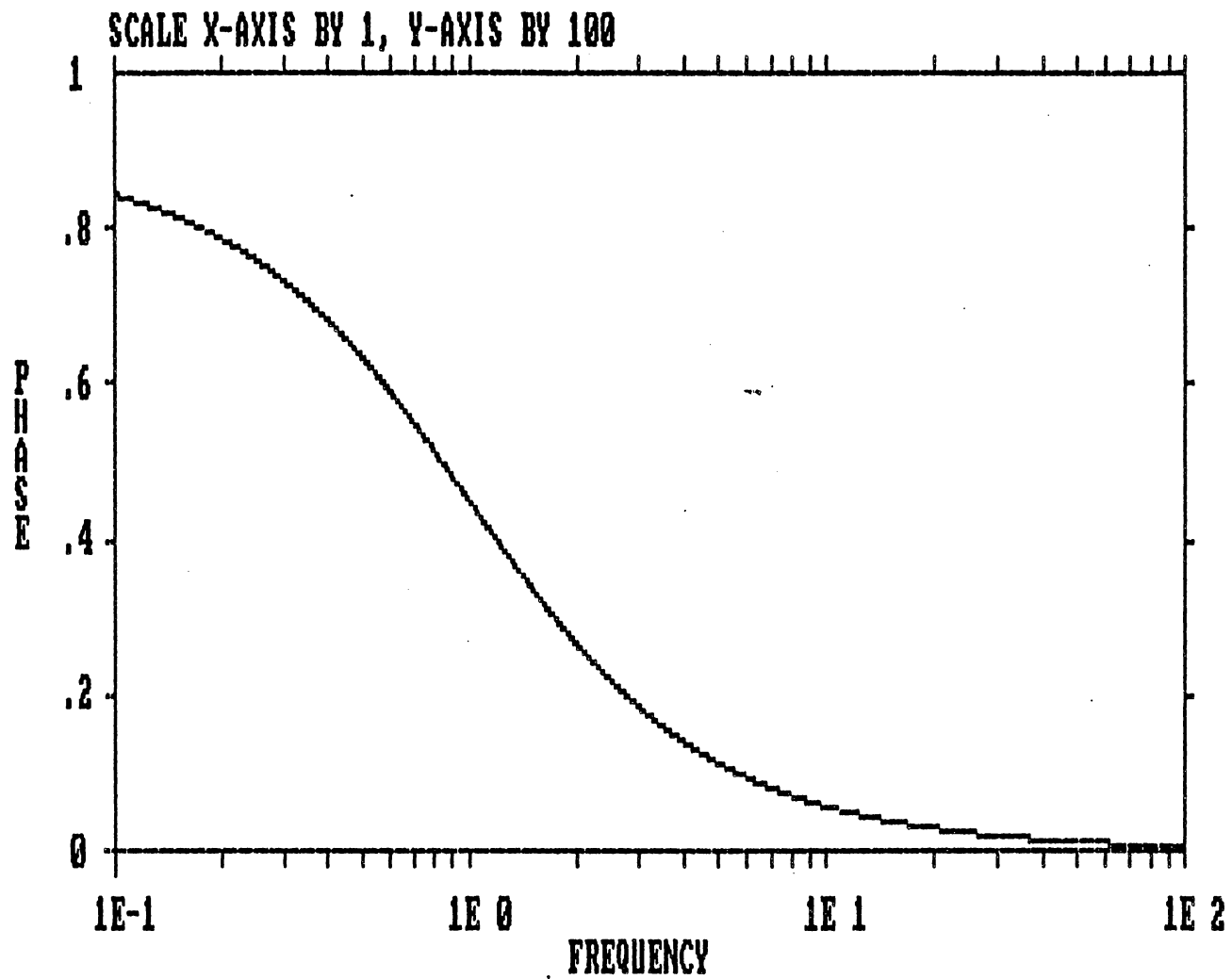


Figure 37. Bode Phase Example

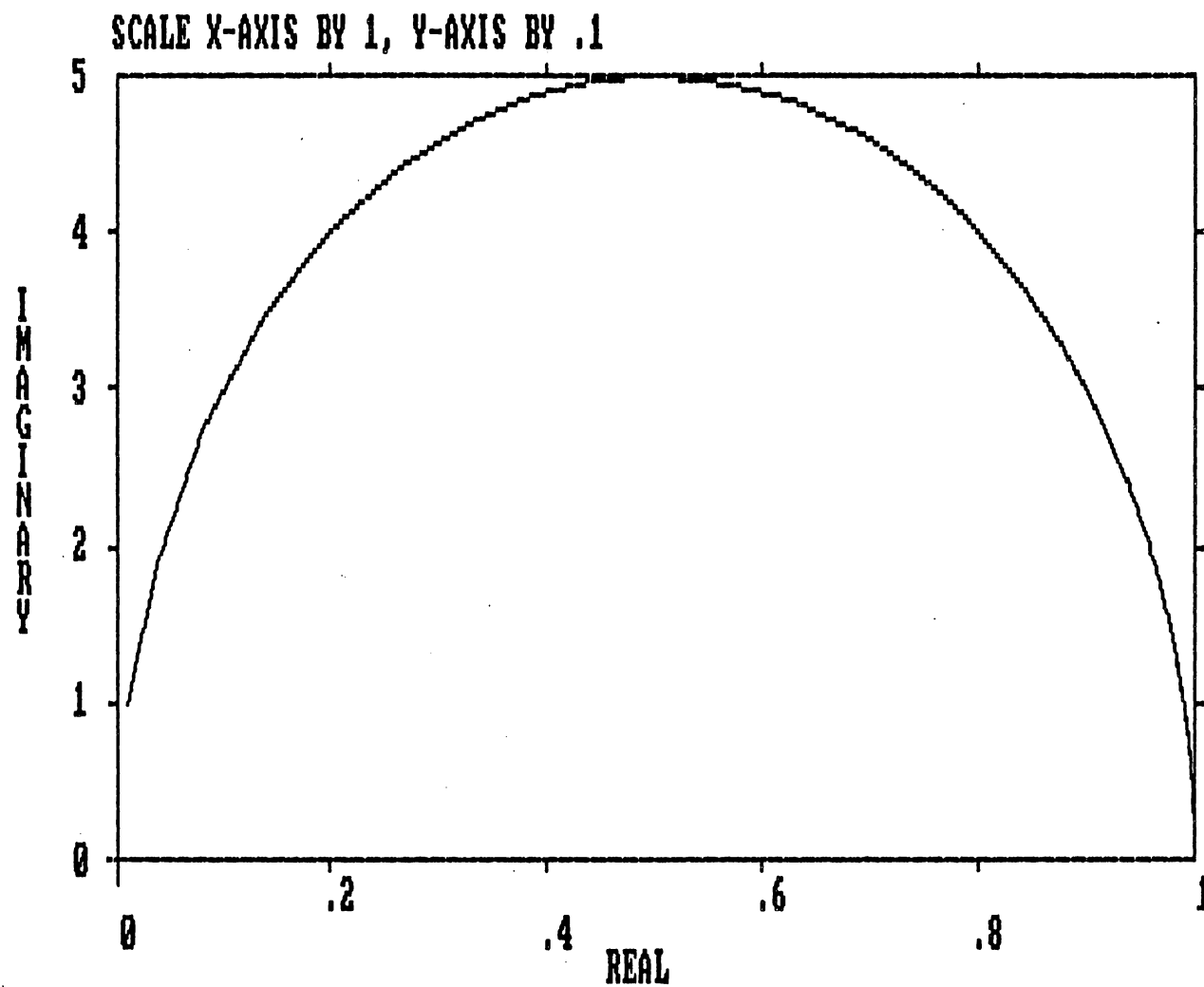


Figure 38. Nyquist Diagram Example

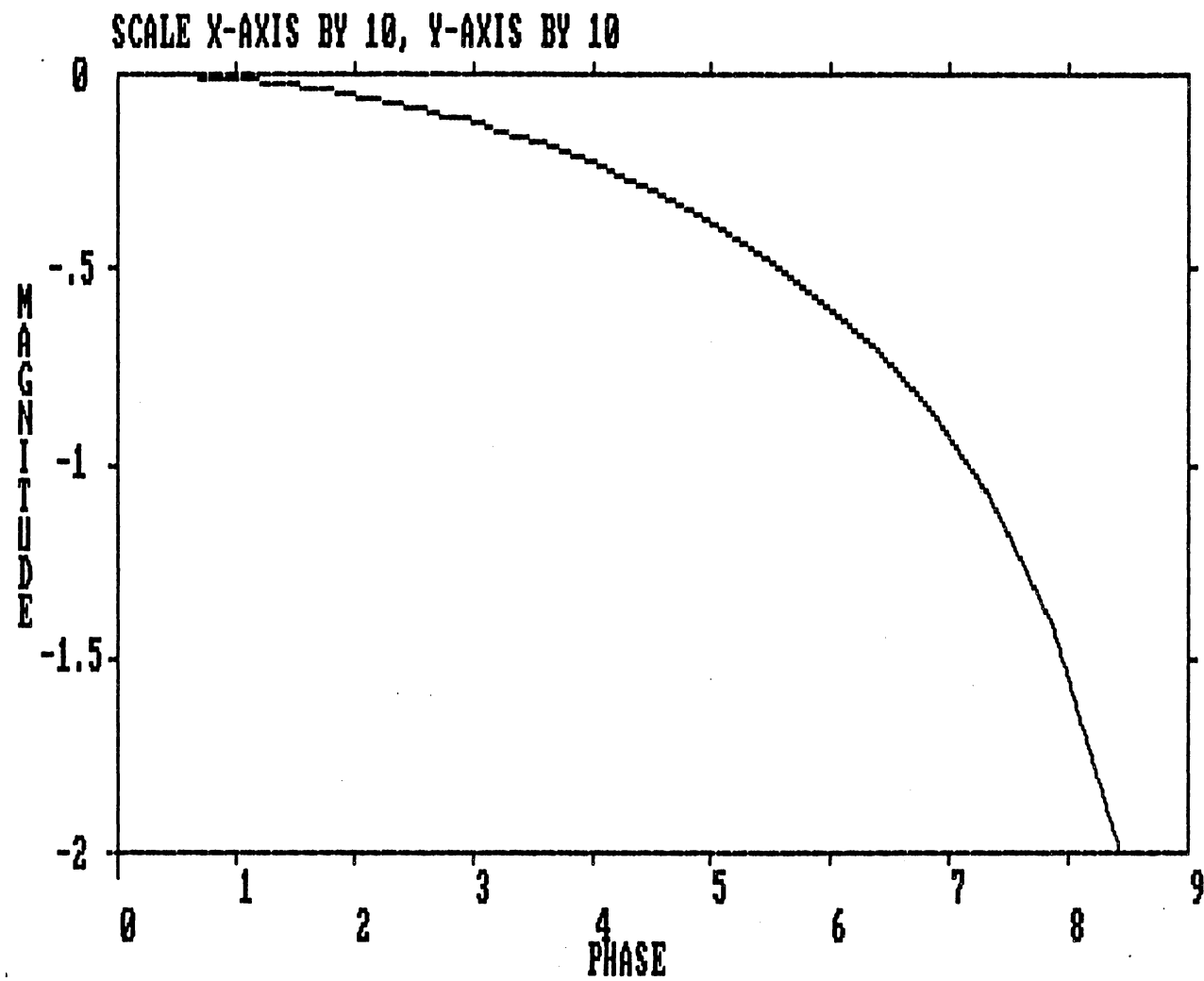


Figure 39. Nichols Chart Example

$$y = x_3 \quad (55)$$

It is desired to determine the system response to a unit step input.

The following information is entered into DACS

SYSTEM DIMENSION >	3
NUMBER OF INPUTS >	1
NUMBER OF OUTPUTS >	1
A MATRIX >	$\begin{bmatrix} 0 & 0 & -1600 \\ 1 & 0 & -100 \\ 0 & 1 & -20 \end{bmatrix}$
B MATRIX >	$\begin{bmatrix} 1600 \\ 0 \\ 0 \end{bmatrix}$
C MATRIX >	$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$
D MATRIX >	$\begin{bmatrix} 0 \end{bmatrix}$
XO MATRIX >	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
MAXIMUM TIME >	10
TIME INTERVAL >	.001
INPUT NUMBER >	1
INPUT TYPE >	STEP
INPUT AMPLITUDE >	1
STARTING TIME >	0
ENDING TIME >	10

Performing calculations and graphing the results generates the display of Fig. 40 which is the system response.

A.4 Time Response

Consider the following system model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -5 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u \quad (56)$$

$$y = x_1 \quad (57)$$

$$u(s) = \frac{10}{s + 5} \quad (58)$$

The closed-form time response of each state x and the output y is desired. The following information is entered into DACS.

```

SYSTEM DIMENSION >      3
A MATRIX >                $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -5 & -5 \end{bmatrix}$ 
B MATRIX >                $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ 
C MATRIX >                $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$ 
K MATRIX >                $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ 
X0 MATRIX >               $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ 

```

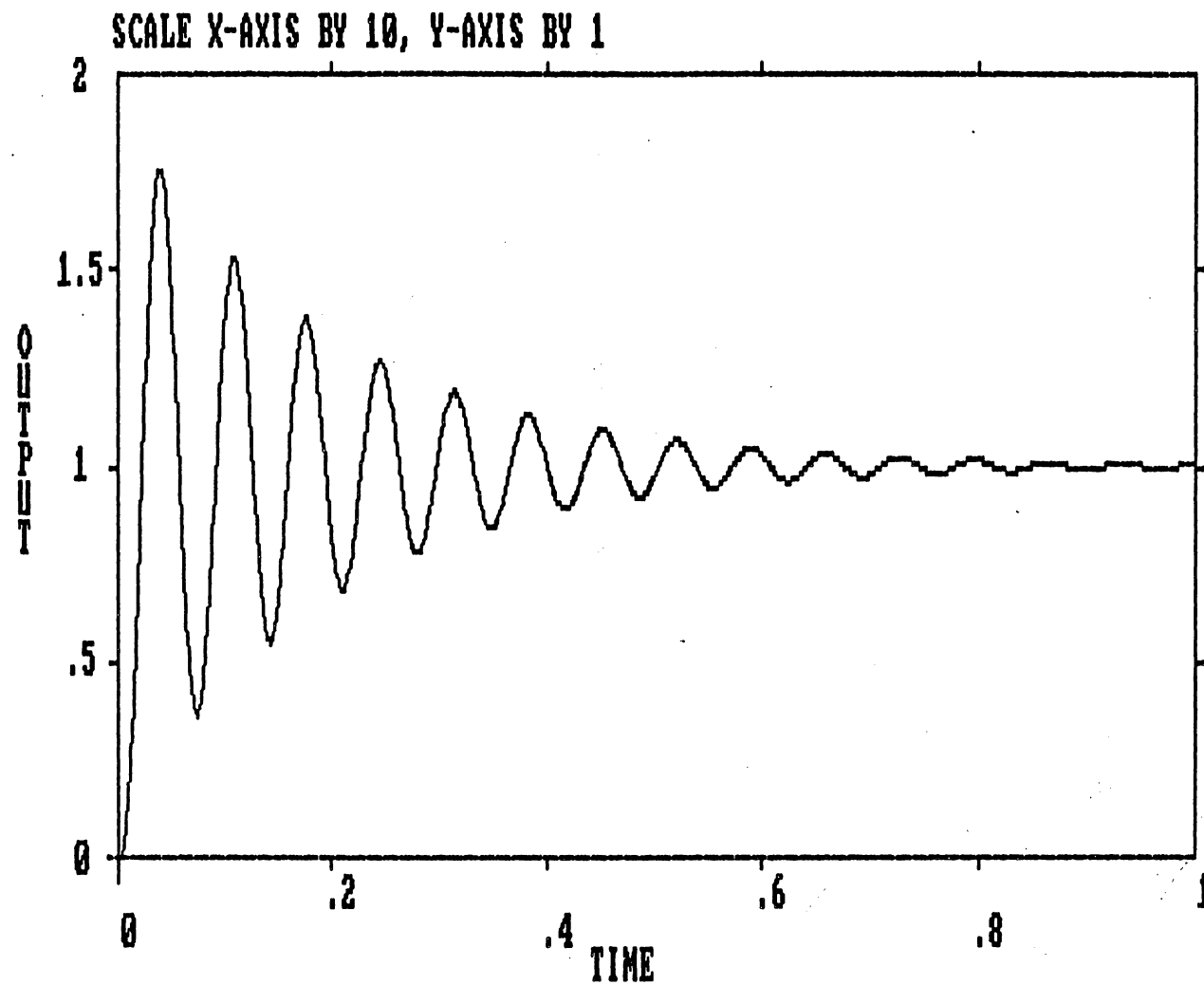


Figure 40. System Simulation Example

```

GAIN KU >          10
GAIN KR >          1
DIMENSION N(s) >    0
DIMENSION D(s) >    1
D(s) POLYNOMIAL >   5

```

After calculations are complete DACS returns the information in Figs. 41 through 45 which are the desired results.

A.5 Matrix Functions

Consider the matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -2 & -3 & -4 \end{bmatrix} \quad (59)$$

It is desired to obtain the determinant, inverse, characteristic polynomial, eigenvalues, state transition matrix, and resolvent matrix associated with matrix A. Entering into DACS the following information

```

MATRIX DIMENSION >    4

```

```

MATRIX >

```

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -2 & -3 & -4 \end{bmatrix}$$

STATE RESPONSE: VECTOR COEFFICIENT OF $\text{EXP}(-4.073948)t$ IS

R 1	.9431759
R 2	-3.842437
R 3	15.65389

Press any key.....

Figure 41. Time Response Example - First Vector

STATE RESPONSE: VECTOR COEFFICIENT OF $\text{EXP}(-5)t$ IS

R	1	-2.500002
R	2	2.500004
R	3	-12.500002

Press any key.....

Figure 42. Time Response Example - Second Vector

STATE RESPONSE: VECTOR COEFFICIENT OF $\text{EXP}(-.463026)t * \text{COS}(1.006438)t$ IS

R 1 .778414
R 2 .671217
R 3 -1.576938

Press any key.....

Figure 43. Time Response Example - Third Vector

STATE RESPONSE: VECTOR COEFFICIENT OF $\text{EXP}(-.463026)t * \text{SIN}(1.006438)t$ IS

R	1	1.556828
R	2	1.342434
R	3	-3.153874

Press any key.....

Figure 44. Time Response Example - Fourth Vector

RESPONSE OF THE OUTPUT $Y(t)$ IS THE SUMMATION

$$\begin{aligned} &.9431759 * \text{EXP}(-4.073948)t \\ &-2.5000002 * \text{EXP}(-5)t \\ &.778144 * \text{EXP}(-.463026)t * \text{COS}(1.006438)t \\ &1.556828 * \text{EXP}(-.463026)t * \text{SIN}(1.006438)t \end{aligned}$$

Press any key.....

Figure. 45. Time Response Example - Output Equation

and for state transition matrix calculations

TIME STEP > 1

the information in Figs. 46 through 54 is obtained.

DETERMINANT = 1

Press any key.....

Figure 46. Matrix Determinant Example

INVERSE

R	1	-2	-3	-4	-1
R	2	1	0	0	0
R	3	0	1	0	0
R	4	0	0	1	0

Press any key.....

Figure 47. Matrix Inverse Example

CALCULATING DATA

The Characteristic Polynomial in ascending powers of s
1 2 3 4 1

Press and key.....

Figure 48. Matrix Characteristic Polynomial Example

The Eigenvalues of the matrix are

	Real Part	Imaginary Part
R 1	- .0467993	.6765268
R 2	-4.679948E-02	-.6765268
R 3	- .6723784	0
R 4	-3.234023	0

Press any key.....

Figure 49. Matrix Eigenvalues Example

STATE TRANSITION MATRIX

R 1	.979482		.9543671		.4284213		6.901623E-02
R 2	-6.901623E-02		.8414495		.7473183		.1523564
R 3	- .1523564	-	.3737291		.3843802		.1378927
R 4	- .1378926	-	.4281415	-	.7874069	-	.1671904

Press any key.....

Figure 50. Matrix State Transition Matrix Example (T = 1)

The Matrix Coefficients of the Numerator of the Resolvent Matrix

MATRIX COEFFICIENT OF S^3

R	1	1	0	0	0
R	2	0	1	0	0
R	3	0	0	1	0
R	4	0	0	0	1

Press any key.....

Figure 51. Matrix Resolvent Example - First Coefficient

The Matrix Coefficients of the Numerator of the Resolvent Matrix

MATRIX COEFFICIENT OF S^2

R 1	4	1	0	0
R 2	0	4	1	0
R 3	0	0	4	1
R 4	-1	-2	-3	0

Press any key.....

Figure 52. Matrix Resolvent Example - Second Coefficient

The Matrix Coefficients of the Numerator of the Resolvent Matrix

MATRIX COEFFICIENT OF S^1

R	1	3	4	1	0
R	2	0	3	4	1
R	3	-1	-2	0	0
R	4	0	-1	-2	0

Press any key.....

Figure 53. Matrix Resolvent Example - Third Coefficient

The Matrix Coefficients of the Numerator of the Resolvent Matrix

MATRIX COEFFICIENT OF S^0

R	1	2	3	4	1
R	2	-1	0	0	0
R	3	0	-1	0	0
R	4	0	0	-1	0

Press any key.....

Figure 54. Matrix Resolvent Example - Fourth Coefficient

APPENDIX B: DACS USER'S GUIDE

The following is the user's guide designed to aid in the usage of DACS.

**DACS: DESIGN AND ANALYSIS
OF LINEAR CONTROL SYSTEMS**

**(C) COPYRIGHT 1984
W. IVAN LEWIS, III**

**USER'S GUIDE
(VERSION 1.2)**

INSTALLATION GUIDE
and
USER'S MANUAL

DACS

A Program for the
Design and
Analysis of linear
Control
Systems

December 1984
(IBM Personal Computer Vers 1.2)

W. Ivan Lewis, III
804 Preston Avenue
Blacksburg, Virginia 24060
703-552-4464

Copyright (C) 1984 W. Ivan Lewis, III
All Rights Reserved

READ THIS FIRST

DACS will enhance the usefulness of your IBM Personal Computer by adding the capability to design and analyze linear control systems. This product operates under PC-DOS, the disk operating system for the IBM Personal Computer. The purchase of DACS does not include the PC-DOS operating system which must be purchased from IBM or an authorized dealer. To run DACS, your system must consist of the following:

- IBM-PC, PC-XT, PC-Portable, or PC-AT Computer System
- 160K bytes RAM
- 1 Diskette drive (2 recommended)
- IBM Color/Graphics Adapter (or equivalent)
- IBM Color/Graphics Monitor (or equivalent or Monochrome graphics monitor)

All of the program modules of DACS may be copied using the standard PC-DOS file copy commands (they are not copy protected). This was chosen for your convenience and does not imply that the program modules or this manual may be freely copied and distributed to others. DACS is the copyrighted property of W. Ivan Lewis, III and only backup copies may be made of the copyrighted material on the distributed diskette and manual.

In the following sections, when a command is to be entered from the keyboard, this manual will show either an exact command or an example. The notation used throughout this manual will show the prompt characters that the computer sends to inform you that a command may be entered, the command that you are to enter (which is underlined), and the character

used to enter the command into the computer. The symbol <cr> is used to indicate the ENTER key which corresponds to the carriage return key on a typewriter or other computer terminal.

Example Command

-----			This symbol is the system prompt that
:			the PC-DOS operating system sends to the
:			screen to indicate that it can accept
:			a command from the user. It also
:			indicates which drive is "active".
:			
:	-----		This is the command to be entered by
:	:		the user. It is either the name of a
:	:		PC-DOS command such as COPY, or the
:	:		name of a command used by DACS
:	:		
:	:	-----	Enter key tells the computer to process
:	:	:	the command that the user has typed.
:	:	:	
:	:	:	
:	:	:	
A>	<u><command></u>	<u><cr></u>	

TABLE OF CONTENTS

	<u>Page</u>
I. Installation of DACS	104
II. Getting Started - General Features of DACS	107
III. Description of DACS Program Modules	110
IV. A Typical Session with DACS	117
V. A Note to DACS Users	127

SECTION I: INSTALLATION OF DACS

This section describes the procedure to follow in order to create a usable copy of DACS and preserve the distributed master disk as a backup. The procedure will be described for an IBM-PC with two diskette drives. If your system has only one drive, please refer to your PC-DOS manual on how to copy files from one diskette to another.

The procedure for creating a DACS diskette involves three steps which are outlined in detail below:

1. Format a new diskette
2. Put the desired system (PC-DOS) utilities on it
3. Put DACS modules on it

You will need a blank diskette (or one which contains information that you no longer need).

1. Place your PC-DOS system disk in drive A (left)
2. Place the diskette to be formatted in drive B
3. Simultaneously press the Ctrl-Alt-Del keys to reset the computer and insure a fresh copy of the PC-DOS operating system
4. Enter the time and data as requested or press <cr>.
5. Enter the following command to format the disk:

A>FORMAT B:/S<cr>

FORMAT will ask you to load the diskette to be formatted in drive B: and strike any key to begin the process. Press <cr> to start formatting your diskette. This formatting process prepares the diskette to accept files and transfers the operating system to the disk. When FORMAT is complete, the program will list the amount of storage available on the new diskette, the number of bad sectors, if any, and ask you if more diskettes are to be formatted. Respond with a "N".

CAUTION: If FORMAT indicates that there are bad sectors on a newly formatted diskette, you should consider using a different diskette.

6. You should include some utility programs from your PC-DOS diskette such as CHKDSK and GRAPHICS on your DACS diskette. Put them on your newly formatted diskette with the command:

A>COPY <filename> B: <cr>

You have now created part of your new system master diskette which is ready for the DACS files. Use the following procedure to create the rest of your DACS master diskette:

1. Remove the PC-DOS master from drive A and put it away.
2. Remove the partially created PC-DOS DACS master and place it in drive A.
3. Put the DACS diskette in drive B.
4. Enter the command: A>COPY B:*.EXE

The files: DACS.EXE, ROOT.EXE, FREQ.EXE, SIM.EXE, RAT.EXE, MAT.EXE and BASRUN.EXE will be copied from drive B to drive A.

5. Remove the DACS distributed diskette from drive B and put it away in a safe place.

This completes the procedure for creating a DACS master diskette. To run DACS simply place your newly created DACS master diskette in drive A and enter the command

A>DACS<cr>

You may wish to run the GRAPHICS.COM program you copied from your PC-DOS diskette before running DACS. To run this program simply enter the command

A>GRAPHICS<cr>

This will load a screen print utility into your computer's memory that will allow you to print graphics display screens on an IBM Personal Computer Graphics Printer or compatible. For further details please refer to your PC-DOS manual.

You may also want to create an AUTOEXEC.BAT file so that DACS will automatically be executed when you boot your system. To create such a file make sure your new DACS disk is in drive A: and enter the following commands:

A>COPY CON AUTOEXEC.BAT<cr>
GRAPHICS<cr>
DACS<cr>
<F6><cr>

where <F6> means press function key 6. Your computer will respond with "1 file(s) copied." The file AUTOEXEC.BAT which consists of two commands, now resides on your DACS diskette. To run DACS simply press Ctrl-Alt-Del or turn your computer off and back on or type AUTOEXEC and hit the ENTER key. When your system boots it will automatically run the AUTOEXEC.BAT file which loads the GRAPHICS print utility and starts DACS.

SECTION II: SETTING STARTED - GENERAL FEATURES OF DACS

DACS consists of the five program modules Root Locus, Frequency Response, System Simulation, Time Response, and Matrix Functions. These program modules provide several methods to design and analyze linear systems. The Root Locus module is used to obtain a plot of the roots of a system as a function of some parameter, usually gain. The Frequency Response module allows you to obtain Bode plots, Nyquist diagrams, and Nichols charts. The System Simulation module simulates systems over time, while the Time Response module calculates the time response of a system in closed-form. The Matrix Functions module is a collection of matrix manipulation programs that allow calculation of determinants, inverses, characteristic polynomials, eigenvalues, and other matrix properties. The state-space description is used in the Simulation and Time Response modules while the system transfer function is used in the Root Locus and Frequency Response modules. Transfer function models may be interchanged between Root Locus and Frequency Response.

The rest of this manual is written to aid you in the use of DACS. This section will describe general program features that are applicable to all of the program modules. The next section contains individual descriptions of each program module and outlines features that are unique to that particular module. Perhaps the most important section, as far as using DACS, is Section IV in which a sample session is given that explains, by example, how to use the program. A brief note to DACS users is given in Section V.

General Features

DACS is a menu-driven program with several features to aid the user. Among these are:

1. Function Key Input: All menu selections make use of the IBM-PC's 10 function keys. These are the gray keys on the left-hand side of the keyboard, arranged in two columns of five keys each. In this manual these keys will be represented by the letter F followed by the key number. For example <F3> would mean to press function key three. Sometimes the function keys will be used in conjunction with either the control key, the shift key, or the alternate key. In this case, you may see <Alt-F3>, which instructs you to hold down the alternate key and press function key three.
2. Help Option: The help option may be toggled on and off from any menu by the key combination <Alt-F1>. When help is on, all required input is explained in detail on the monitor. Once you are familiar with **DACS**, you may wish to turn off the help option to expedite parameter input. The status of the help option is displayed on each menu.
3. Sound: Three levels of sound are provided. Sound level 2 provides input tones when a keyboard entry is required and error tones for incorrect input. Sound level 1 provides error tones only, while Sound level 0 disables sound completely. Sound level may be changed with the key combination <Alt-F2> and the status of the sound level is displayed on each menu.
4. Session Archiving: When the archiving option is activated an archiving file is kept by the program. In this file are stored a record of all menu selections chosen and all filenames used. In addition, comments may be entered into the archive at any time. This file provides a complete record of your session with **DACS**. Archiving is activated by the key combination <Alt-F3>. Once activated, pressing <Alt-F3> again allows comments to be entered into the archive.
5. Directory Information: You may obtain a listing of the files on a disk from any main menu by using <F9>. In addition, each time you save or recall data from disk, both file directory and filename information may be obtained. Unwanted files may be erased at this time.
6. Escape Capability: The escape key always provides an exit to the next higher command level.

7. Automatic Scaling: All graphs are originally scaled by DACS.
8. Zooming: Once a graph is displayed, you may zoom in or out as often as desired. To zoom, use the key combination <Ctrl-F5>. You will be prompted for the minimum and maximum values for the x and y axes of the new graph. (Note: this option is not available on Bode plots when both magnitude and phase are plotted on the same screen, or on graphs recalled from disk files.)
9. Cursor Control: When a graph is displayed, you may move the cursor to any point within the boundaries of the graph and get coordinate information about that point. Simply use the appropriate cursor keys to locate the cursor at the desired position and press return. By holding down the shift key while using the cursor keys, the cursor will proceed at a faster rate. (Note: This option is available under the same conditions as the zooming option above.)
10. Screen Dumps: Any displayed graph may be saved in a file on disk and redisplayed at a later time. To save a graph, use the key combination <Ctrl-F4> while the graph is displayed. To redisplay a disk file, use the key combination <Ctrl-F2> when in the GRAPH DATA option. (Note: The GRAPHICS screen print utility must be executed before running DACS to enable screen dumps. For more information refer to SECTION I.)
11. Grids: Graphs may be displayed with or without grids. You will be asked if grids are desired before each graph. (Note: Grids are not available on Bode plots.)
12. Default Answers: There are several times when DACS will ask you a yes or no question. To answer, simply press the <Y> key for yes or the <N> key for no. The default answer is always no so simply hitting the <cr> is equivalent to a no answer. (Note: DACS will accept both upper case and lower case letters for yes and no.)
13. Erasur Checks: There are several checks built into DACS to insure that information is not lost. For example, if you already have entered a model into the program and you select the ENTER NEW MODEL options, DACS will ask you if you want to erase the existing model. A yes answer is required before any action is taken.

SECTION III: DESCRIPTION OF DACS PROGRAM MODULES

This section contains a brief description of the Root Locus, Frequency Response, System Simulation, Time Response, and Matrix Functions modules. Features unique to each module are described.

Root Locus

This module may be used to obtain a plot of the location of the roots of a system as a function of some parameter, usually gain. The system is described by its open-loop transfer function as

$$G(s) = K \frac{N(s)}{D(s)} \quad (60)$$

where K is the independent parameter. The problem is solved by finding the roots to the system characteristic equation

$$1 + G(s) = 0 \quad (61)$$

for each value of K and then plotting the results. All systems of fourth order and less are solved in complete closed-form to expedite solution and a modified-Bairstow algorithm is used to find the roots of systems of higher order.

Frequency Response

This module is used when a Bode plot, Nyquist Diagram, or Nichols chart is desired. As with the Root Locus module, the system is

described by its open-loop transfer function, Eq. 60. In order to obtain true phase information for the Bode and Nichols plots, the numerator and denominator polynomials are first factored. The amplitude ratio in decibels is then merely the product of the individual magnitudes of the factors and the phase angle is the sum of the angles associated with the individual factors. A minus sign is associated with angles of the denominator terms.

Nyquist calculations simply amount to evaluating the transfer function at a complex number, in this case the frequency of interest. The real and imaginary parts of each calculated number then become the x and y coordinates of a point on the Nyquist diagram.

All parameter input is the same regardless of the type of plot desired. When you instruct DACS to calculate data you will be prompted with

```
<Sft-F7>....BODE CALCULATIONS  
<Sft-F8>....NYQUIST CALCULATIONS  
<Sft-F9>....NICHOLS CALCULATIONS
```

It is at this point that you instruct DACS which type of plot you want by pressing the appropriate key combination. DACS remembers your request so that it can generate the appropriate graph after calculations are complete.

When Bode calculations are performed, two curves are generated, magnitude versus frequency and phase versus frequency. These two plots may be displayed simultaneously on the screen or separately. When the data is plotted DACS will ask if they should be plotted separately.

Answer <N> to obtain both on one screen and <Y> to obtain individual plots.

When Nichols calculations are performed, the resulting curve is a plot of magnitude versus phase and the frequency variable is eliminated. When a Nichols chart is displayed, however, you may move the cursor to a point on the curve and obtain open-loop magnitude and phase, closed-loop magnitude and phase, and frequency information about that point. Simply move the cursor as described in SECTION I to the desired point and press <cr> to obtain open-loop magnitude and phase. Pressing any key will change the coordinate information displayed to closed-loop magnitude and phase. Pressing any key again will display the frequency of that point. The closed-loop information is obtained from the calculated open-loop data by the two formulas

$$M = \frac{G}{(G^2 + 2G\cos \theta + 1)^{1/2}} \quad (62)$$

and

$$\tan \phi = \frac{\sin \theta}{G + \cos \theta} \quad (63)$$

where M = closed-loop magnitude
 G = open-loop magnitude
 ϕ = closed-loop phase
 θ = closed-loop phase

(Note: The Nichols calculations use unity feedback for closed-loop information, that is, $H=1$.) Frequency information is obtained by a table search of the calculated data.

When Nyquist calculations are performed, the resulting curve is a polar plot of the complex function

$$G(j\omega_f) = K \frac{N(j\omega_f)}{D(j\omega_f)} \quad (64)$$

as the input frequency ω_f is varied throughout the specified range.

System Simulation

This module can be used to determine the time-domain response of a system. The system is described by a state model of the form

$$\dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(t) \quad (65)$$

$$\underline{y}(t) = C\underline{x}(t) + D\underline{u}(t) \quad (66)$$

where

$\underline{x}(t)$ = vector of system states
 $\underline{u}(t)$ = vector of system inputs
 $\underline{y}(t)$ = output vector

and A, B, C, and D are constant matrices. It can be shown that the solution to this set of equations is

$$\underline{x}(t) = e^{AT} \underline{x}(0) + \int_0^t e^{A(t-\tau)} B \underline{u}(\tau) d\tau. \quad (67)$$

The matrix exponential, e^{AT} , is called the state transition matrix and is calculated by the truncated infinite series

$$e^{AT} = I + AT + \frac{1}{2!} A^2 t^2 + \frac{1}{3!} A^3 t^3 + \dots + \frac{1}{K!} A^K t^K \quad (68)$$

If the input, $u(t)$ to the system is assumed constant over the time interval then it may be removed from the integral making solution much easier. For time varying inputs, it is important to pick a time interval over which the input varies relatively little (the definition of "relatively little" is left up to you).

IMPORTANT: FOR TIME VARYING INPUTS, IT IS IMPORTANT TO PICK A TIME INTERVAL OVER WHICH THE INPUT VARIES RELATIVELY LITTLE.

Up to five external inputs may be applied to the system at one time. When graphing the calculated data for multiple output systems, **DACS** will ask you which output you want for the abscissa. You may press <cr> and the abscissa will default to time or you may choose the other outputs to be graphed as a function of an output of your choosing. **DACS** will then ask which outputs should be graphed. You may press <cr> and all outputs will be plotted or you may select outputs you desire.

Time Response

The Time Response module may be used to calculate the time response in closed-form of the system.

$$\dot{\underline{x}}(t) = A\underline{x}(t) + \underline{b}u(t) \quad (69)$$

$$y(t) = \underline{c}^T \underline{x}(t) \quad (70)$$

where

$\underline{x}(t)$ = vector of system states

$\underline{y}(t)$ = single output

and

$u(t)$ = single input.

The input $u(t)$ is the combination of a feedback of system states and an externally applied input $r(t)$, such that

$$u(t) = K[r(t) - \underline{k}^T \underline{x}(t)]. \quad (71)$$

The externally applied input is described by a system transfer function of the form

$$R(s) = K_r \frac{N(s)}{D(s)}. \quad (72)$$

The solution method is to combine the input with the system to form a homogeneous augmented system of the form

$$\dot{\underline{x}}(t) = A_a \underline{x}(t). \quad (73)$$

The solution to this homogeneous system then becomes

$$\underline{x}(t) = e^{A_a t} \underline{x}(0). \quad (74)$$

The state transition matrix is calculated using Sylvester's expansion theorem which states that

$$e^{A_a t} = F_1 e^{\lambda_1 t} + F_2 e^{\lambda_2 t} + \dots + F_m e^{\lambda_m t} \quad (75)$$

where the λ_i 's are the eigenvalues of the matrix A_a . In this way the

system states and the system output may be expressed as equations in terms of the eigenvalues of the original system and the applied input. The algorithm used to evaluate the F matrices is not valid for systems which have repeated eigenvalues. Should your system fall into this category DACS will inform you.

A unique feature to this program module is the ability to save and recall both the system model and the system input.

Matrix Functions

The matrix functions module is simply a collection of matrix manipulation programs. Within this module you may calculate the determinant, inverse, characteristic polynomial, eigenvalues, state transition matrix and resolvent matrix of a given matrix. You will be presented with a menu listing these options when you select the CALCULATE DATA menu option.

SECTION IV: A TYPICAL SESSION WITH DACS

Consider the system with the open-loop transfer function

$$G(s) = K \frac{s + 1.2}{s^3 + 9s^2 + 8s} \quad (76)$$

Let us use **DACS** to plot the root locus of this system for values of gain from 0 to 30.

Place your **DACS** diskette in drive A and turn on your PC. If your PC is already on, press Ctrl-Alt-Del to reboot the system. If you created an AUTOEXEC.BAT file in SECTION I then **DACS** will run automatically, otherwise type

A>DACS <cr>

to run the program. Once **DACS** is loaded, you will be greeted with the message

**DACS: DESIGN AND ANALYSIS
OF LINEAR CONTROL SYSTEMS**

**(C) Copyright 1984
W. Ivan Lewis, III**

Press Space Bar to Continue

and your PC will beep, indicating that you are required to enter a command from the keyboard. Press the space bar and **DACS** will present

you with the main menu.

CONTROLS PROGRAMS

```

<F1>.....ROOT LOCUS
<F2>.....FREQUENCY RESPONSE
<F3>.....SYSTEM SIMULATION
<F4>.....TIME RESPONSE
<F5>.....MATRIX FUNCTIONS
<F9>.....FILE DIRECTORY
<ESC>.....EXIT TO SYSTEM
<Alt-F1>.....HELP LEVEL OFF
<Alt-F2>.....SOUND LEVEL 1
<Alt-F3>.....ARCHIVE/COMMENTS

```

PRESS DESIRED FUNCTION KEY

You have ten options at this point, each of which is listed on the menu. Since this is your first session with DACS, turn on the help option by holding down the Alternate key and pressing function key one. Notice that the menu is updated and now indicates that help has been turned on. The default sound level is 1, which, as explained in SECTION II, provides you with error signals only. Change to sound level 2 by holding down the Alternate key and pressing function key two. The computer beeps and the menu is updated again. Sound level 2 provides you with both error signals and input prompt signals. Notice that the PC beeped indicating that it is waiting for you to make another selection.

Since we are interested in a root locus plot of our system, press <F1>. The disk drive light will come on indicating that the root locus module is being loaded and you will be presented with a new menu.

ROOT LOCUS

```

<F1>.....ENTER NEW MODEL
<F2>.....RECALL OLD MODEL
<F3>.....MODIFY CURRENT MODEL
<F4>.....STORE MODEL ON DISK
<F5>.....ENTER INPUT PARAMETERS
<F6>.....CALCULATE DATA
<F7>.....GRAPH DATA
<F8>.....LIST/SAVE DATA
<F9>.....FILE DIRECTORY
<ESC>.....EXIT TO MAIN MENU
<Alt-F1>.....HELP LEVEL ON
<Alt-F2>.....SOUND LEVEL 2
<Alt-F3>.....ARCHIVE/COMMENTS

```

PRESS DESIRED FUNCTION KEY

The first order of business is to describe the system of interest to DACS. Press <F1> so that you may enter the system. Since the help option is turned on, DACS provides you with a help screen to aid in model entry. The screen appears as

The Root Locus program calculates the zeros of the equation

$$1 + G(s)$$

as a function of K. $G(s)$ is assumed rational and

$$G(s) = KN(s)/D(s)$$

You will be prompted for the polynomials $N(s)$ & $D(s)$. The coefficient of the highest order term is assumed unity. Enter the coefficient in ascending order. Examples:

$$G(s) = (s + 3)/(s^3 + 2s^2 + 9s + 3)$$

$$N(s) = 3$$

$$D(s) = 3 \quad 9 \quad 2$$

The coefficient of the highest order term is not entered.

Press any key

Therefore, all that is required to describe the system to DACS is to enter the coefficients of each polynomial in ascending order. Continue by pressing a key. The first two pieces of information that the program needs are the dimensions of the numerator and denominator of our system. DACS requests this information by printing

N(s) DIMENSION >

Since our numerator is of first order, respond as shown below.

N(s) DIMENSION > 1 <cr>

You will next be asked for the dimension of the denominator. Respond as shown.

D(s) DIMENSION > 3 <cr>

DACS now prompts you to enter the coefficients of the N(s) polynomial. You are asked to press the <ENTER> key after each complete number. The <ENTER> key is the same as the <cr>. Since our numerator polynomial is only first order, we have only one number to enter. (Remember that the coefficient of the highest order term is assumed unity and is not entered). Type in the number 1.2 and press the <ENTER> key. DACS realizes that this is the only entry for the numerator polynomial and so you are requested to enter the coefficients of the denominator. For this case, you should type in 0<cr>, 8<cr>, and 9<cr>, since these are

the coefficients of the denominator in ascending order. Before cleared, the screen should look like this

ENTER POLYNOMIALS IN ASCENDING ORDER PRESSING <ENTER> AFTER EACH ELEMENT

N(s) POLYNOMIAL

R 1 1.2

D(s) POLYNOMIAL

R 1 0 8 9

If for any reason you make a mistake and want to start over just press <Esc> and, after a message, you will be returned to the root locus menu.

Now you are back at the root locus menu. Your next step is to tell **DACS** the range of gains you want data calculated for. This done under the **ENTER INPUT PARAMETERS** menu option so press <F5>. Again, since help is on, you are presented with some useful information.

You will be asked to enter the minimum gain, **KMIN**, the maximum gain, **KMAX**, and the gain interval, **KINT**. The number of points to be calculated equals

$$(KMAX - KMIN) / KINT.$$

This number may not exceed 500.

Press any key.....

DACS needs to know the range of gains and the step size. Press a key to continue. We know our minimum gain is 0 and our maximum gain is 30 from the problem statement. Let us choose the interval to be 0.1. Enter the parameters as prompted so that your screen appears as

MINIMUM GAIN > 0

MAXIMUM GAIN > 30

GAIN INTERVAL > .1

Again, if you make a mistake and wish to start over just press <Esc>. The system is now completely described and we are ready to have the data calculated. Press <F6> and DACS responds with

CALCULATING DATA.....

This example takes approximately 30 seconds to calculate. When DACS is finished you will be returned to the main menu. You may now graph the data by pressing <F7>. This brings you to the menu shown below.

OPTIONS

<Ctrl-F1>....GRAPH CURRENT DATA
 <Ctrl-F2>....RE-DISPLAY SCREEN FILE
 <ESC>.....END

WHILE GRAPH DISPLAYED

<ENTER>.....CURSOR COORDINATES
 <Ctrl-F4>....SAVE GRAPH ON DISK
 <Ctrl-F5>....ZOOM
 <ESC>.....END

(EXCEPT ON RE-DISPLAY)

Since you are interested in graphing the data just calculated, press <Ctrl-F1>. You will be asked if you want grids on the graph, respond <N> or press <cr>. DACS will then graph the root locus of our model (see Fig. 55).

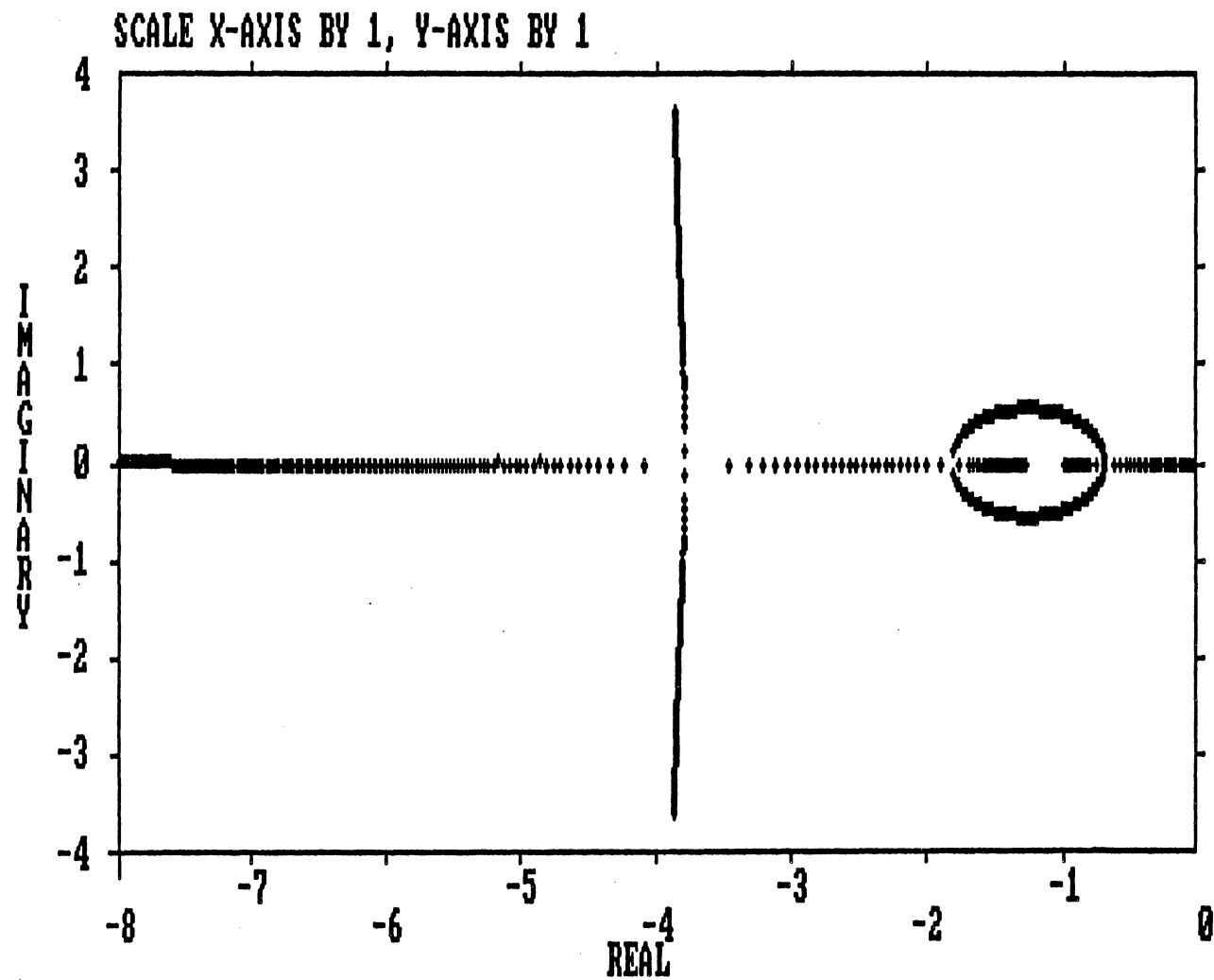


Figure 55. Example Root Locus

Once the graph is displayed you have several options. Use the cursor keys to move the cursor at the origin to any point. (By holding down the shift key while using the cursor keys, the cursor will proceed at a faster rate.) Once at that point, press <cr>. The x and y coordinates of that point are displayed on the upper right portion of the screen. If your PC is connected to a graphic printer and you ran the GRAPHICS.COM file before running DACS then you may hold down the shift key and press the key labeled PrtSc to obtain a hardcopy of this graph. (Figure 55 was produced in this manner.) If you wish to save this graph on a disk then you may press <Ctrl-F4>. You will be prompted for the necessary information. Another option is to zoom in on a portion of the graph. Let us zoom in on the circular portion of the locus. Press <Ctrl-F5>. In the upper right hand portion of the screen you will be prompted for the minimum and maximum x and y coordinates of the new graph. The first prompt is for the minimum value. Let us choose -2. Therefore, enter the following

X1 = -2<cr>

The next prompt is for the maximum x value. Enter 0 <cr>. In the same manner, you will be asked for the minimum and maximum values for the y axis. Enter -1 for the minimum and 1 for the maximum. DACS will respond by redrawing the locus over the requested range. (See Fig. 56.) When you are finished zooming, saving, and inspecting the coordinates of your data, press <Esc> to return you to the graphing

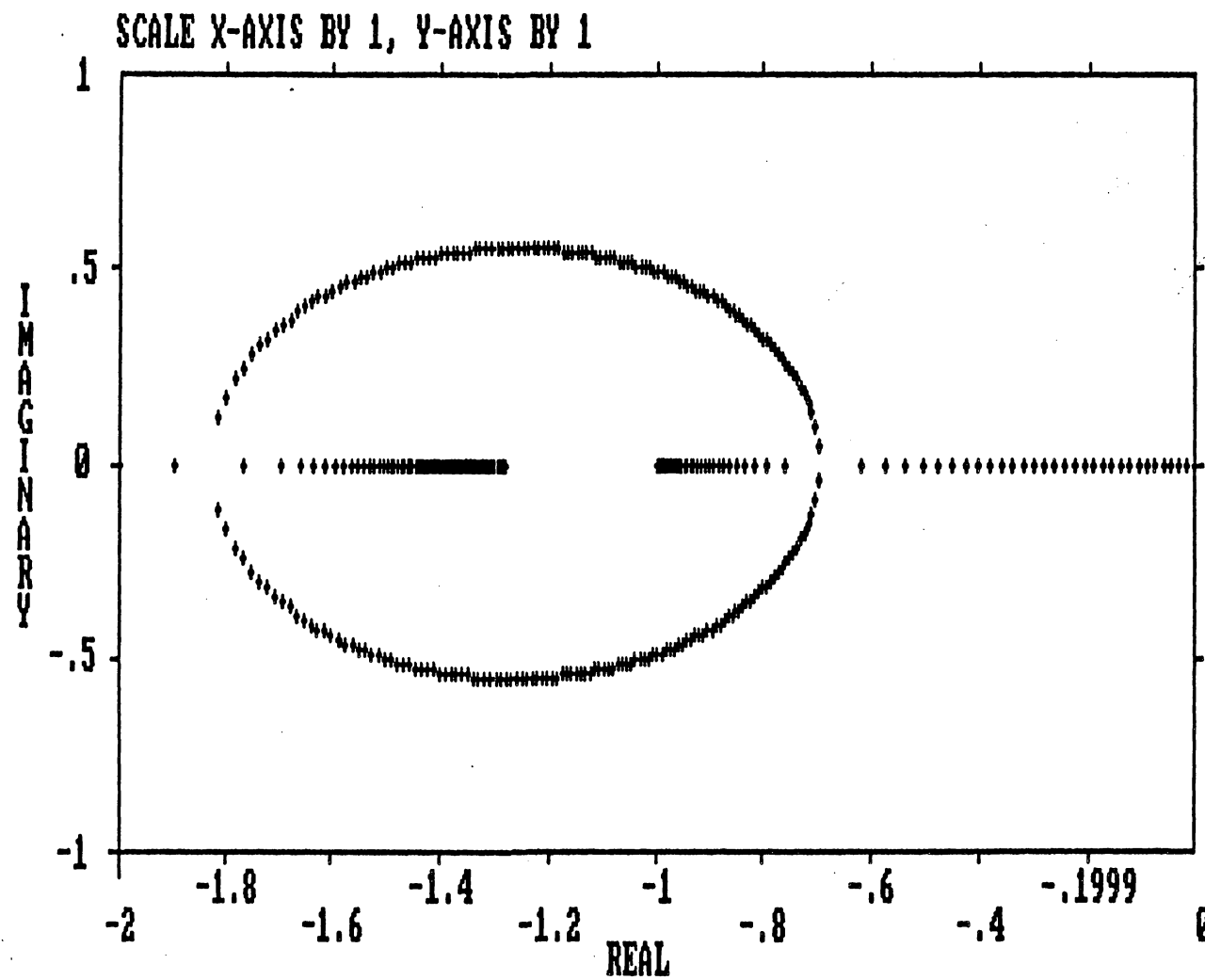


Figure 56. Zoom on Example Root Locus

menu. Press <Esc> again to return you to the root locus menu.

You have just completed your first session with DACS. At this point you may return to the main menu by pressing <Esc> but one word of caution before you do. Once you leave root locus, or any module, you lose the model and data. If you wish to save the model so that you can re-analyze it later, you must save it before leaving root locus.

CAUTION: ONCE YOU LEAVE A PROGRAM MODULE, YOU LOSE ANY MODEL OR DATA ENTERED OR CALCULATED IN THAT MODULE. IF YOU WISH TO SAVE THE MODEL OR DATA, YOU MUST DO SO BEFORE LEAVING THE MODULE.

Press <Esc> to return to the main menu. Press <Esc> again to return you to the operating system.

SECTION V: A NOTE TO DACS USERS

DACS is a useful research, teaching, and design tool for linear control systems, and brings powerful computational ability to your IBM-PC. All effort has been made to make DACS easy to use and error free. Nevertheless, there is always room for improvement. Should DACS fail to perform properly or if you have any comments as to how it may be improved, please fill out the following form and return it to W. Ivan Lewis, III at the address at the beginning of this manual.

DACS USER COMMENT FORM

Please fill out the following form if DACS does not function properly or if you have comments as to how DACS may be improved.

DACS Program Disk Number.

DACS Program Module in use when error occurred.

```
(root, freq, sim, rat, mat)
```

DACS Menu Selection executing when error occurred.

(enter model, recall model, etc.)

DACS Error Description of Comment:

[illegible]

APPENDIX C: DACS PROGRAM LISTINGS

The following is the listing of all programs that comprise DACS. The listing labeled DACS is the main calling routine that interfaces with all the other modules. The listing labeled ROOT is the Root Locus module listing. The listing labeled FREQ is the Frequency Response module listing. The listing labeled SIM is the System Simulation module listing. The listing labeled RAT is the Time Response module listing. Finally, the listing labeled MAT is the Matrix Functions module listing.

C.1 DACS 3/6/85

```

10 COMMON HELP%,HELP$,ARC%,SND%,QW,ARCNAM$:DEFINT I,J:RT$='RETRY?
(Y/N)"
15 IF QW>0 THEN 80 ELSE SCREEN 0,1:CLS:WIDTH 40:COLOR 10,0
20 LOCATE 4,6:PRINT CHR$(213)+STRING$(27,205)+CHR$(184)
25 LOCATE 5,6,0:PRINT CHR$(179)+" DACS: DESIGN AND ANALYSIS "+CHR$(
179)
30 LOCATE 6,6,0:PRINT CHR$(179)+STRING$(27,32)+CHR$(179)
35 LOCATE 7,6,0:PRINT CHR$(179)+" OF LINEAR CONTROL SYSTEMS "+CHR$(
179)
40 LOCATE 8,6,0:PRINT CHR$(212)+STRING$(27,205)+CHR$(190)
45 COLOR 15,0:LOCATE 13,11,0:PRINT"(C)Copyright 1984."
50 LOCATE 14,11,0:PRINT"W. Ivan Lewis, III":COLOR 14,0
55 LOCATE 23,7,0:PRINT"Press Space Bar to Continue":BEEP
60 HELP%=0:HELP$='OFF':ARC%=0:SND%=1:GOSUB 30030:GOSUB 30810
80 QW=1:SCREEN 0,1:CLS:WIDTH 80:COLOR 14,0:LOCATE 6,29:PRINT"CONTRO
L PROGRAMS"
90 COLOR 10,0:LOCATE 9,23,0:PRINT"<F1>.....ROOT LOCUS"
100 LOCATE 10,23,0:PRINT"<F2>.....FREQUENCY RESPONSE"
110 LOCATE 11,23,0:PRINT"<F3>.....SYSTEM SIMULATION"
120 LOCATE 12,23,0:PRINT"<F4>.....TIME RESPONSE"
130 LOCATE 13,23,0:PRINT"<F5>.....MATRIX FUNCTIONS"
140 LOCATE 14,23,0:PRINT"<F9>.....FILE DIRECTORY"
150 LOCATE 15,23,0:PRINT"<ESC>.....EXIT TO SYSTEM"
160 LOCATE 16,23,0:PRINT"<Alt-F1>....HELP LEVEL"
170 LOCATE 17,23,0:PRINT"<Alt-F2>....SOUND LEVEL"
180 COLOR 14,0:LOCATE 16,46,0:PRINT HELP$:LOCATE 17,46,0:PRINT SND%
190 COLOR 10,0:LOCATE 18,23,0:PRINT"<Alt-F3>....ARCHIVE/COMMENTS"
200 COLOR 14,0:LOCATE 21,24,0:PRINT"PRESS DESIRED FUNCTION KEY"
210 INFLG=2:GOSUB 7120:IF ESCK=1 THEN 390
220 IF C=59 THEN CHAIN"ROOT"
230 IF C=60 THEN CHAIN"FREQ"
240 IF C=61 THEN CHAIN"SIM"
250 IF C=62 THEN CHAIN"RAT"
260 IF C=63 THEN CHAIN"MAT"
270 IF C=67 THEN GOSUB 30190:GOTO 80
280 IF C=104 THEN GOSUB 590:GOTO 180
290 IF C=105 THEN GOSUB 550:GOTO 180
300 IF C=106 THEN GOSUB 420:GOTO 80
310 GOSUB 30760:GOTO 210
370 '
380 'RETURN TO SYSTEM
390 CLEAR:WIDTH 80:SCREEN 0,0:COLOR 7,0:CLOSE:END
400 '
410 'ARCHIVE SESSION / ADD COMMENTS
420 COLOR 12,0:CLS:IF ARC%=1 THEN 470 ELSE LOCATE 2,27,0
430 PRINT"ARCHIVE SESSION? (Y/N)":INFLG=1:GOSUB 7120:IF ESCK=1 OR
C$="N" THEN RETURN
440 FILETPE=0:IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
450 FILEXT$='.ARC':FILENUM=2:GOSUB 30650:CLOSE #FILENUM

```

```

460 IF ESCK=1 OR FERR$<>"OK" THEN RETURN ELSE ARCNAM$=FILENAM$:ARC%
=1:CLS
470 COLOR 12,0:LOCATE 2,28,0:PRINT"ENTER COMMENTS? (Y/N)":INFLG=1:
GOSUB 7120
480 IF C$="N" OR ESCK=1 THEN RETURN ELSE NUMDIG=78:INFLG=5:XFLD=1
490 CLS:LOCATE 2,19,0:YFLD=4
500 PRINT"ENTER COMMENTS BELOW....PRESS ESC TO QUIT":PRINT
510 GOSUB 7110:IF ESCK=1 THEN GOSUB 31710:RETURN
520 COM$=S$:GOSUB 30410:IF FERR$<>"OK" THEN RETURN ELSE YFLD=YFLD+1
:XFLD=1:IF YFLD>24 THEN 490 ELSE 510
530 '
540 'SOUND LEVEL
550 SND%=SND%+1:IF SND%>2 THEN SND%=0
560 RETURN
570 '
580 'SET HELP LEVEL
590 IF HELP%=1 THEN HELP%=0:HELP$="OFF":RETURN
600 HELP%=1:HELP$="ON ":RETURN
610 '
7040 'INPUT ROUTINE for numbers, names, function keys, comments, &
Y/N answers
7050 'INFLG=1: gets Y/N answer, returns c$="Y" or c$="N"
7060 'INFLG=2: gets function key, returns c=asc(2nd char of key)
7070 'INFLG=3: gets number as string, checks if valid, returns s=n
umber
7080 'INFLG=4: gets filename, returns s$=filename
7090 'INFLG=5: gets comment from user for archive, returns s$=comm
ent
7100 'Calling routine specifies NUMDIG, XFLD, YFLD, and INFLG
7110 COLOR 6,0
7120 SXFLD=XFLD:S$="":C$="":T$="":N$="":ESCK=0:GOSUB 30030:IF SND%>
1 THEN BEEP
7130 IF INFLG>2 THEN 7150
7140 C$=INKEY$:IF C$="" THEN 7140 ELSE 7170
7150 LOCATE YFLD,XFLD,1
7160 C$=INKEY$:IF C$="" THEN 7150
7170 IF C$=CHR$(27) THEN ESCK=1:RETURN 'escape key
7180 IF INFLG=2 THEN 7370
7190 IF LEN(C$)>1 THEN 7450 'invalid
7200 C=ASC(C$)
7210 IF INFLG=1 THEN 7320
7220 IF C=8 THEN 7500
7230 IF INFLG=5 THEN 7800 'comment
7240 IF INFLG=3 THEN 7390 'number
7250 'Name Routine
7260 IF C=13 THEN RETURN
7270 IF C=46 AND NUMDIG>10 THEN 7470 'okay
7280 IF C=91 OR C=92 OR C=93 OR C=124 THEN 7450 'invalid
7290 IF C=34 OR C=39 OR C=42 OR C=43 OR C=46 OR C=47 THEN 7450 'in
valid
7300 IF (C>32 AND C<59) OR (C>63 AND C<127) THEN 7470 ELSE 7450

```

```

7310 'Yes/No routine
7320 IF C=89 OR C=121 THEN C$="Y":RETURN
7330 IF C=78 OR C=110 THEN C$="N":RETURN
7340 IF C=13 THEN C$="N":RETURN
7350 GOTO 7450
7360 'Function Key
7370 IF LEN(C$)<2 THEN 7450 ELSE C$=RIGHT$(C$,1):C=ASC(C$)
7380 IF (C>58 AND C<69) OR (C>83 AND C<114) THEN RETURN ELSE 7450
7390 'Number Routine
7400 IF C=13 THEN 7540 'carriage return
7410 IF C>47 AND C<58 THEN 7470 'okay
7420 IF C=46 OR C=43 OR C=45 OR C=32 THEN 7470 'okay
7430 IF C=69 OR C=101 THEN C$=CHR$(69):GOTO 7470 'okay
7440 'Illegal character
7450 GOSUB 30760:GOTO 7130
7460 'Character okay
7470 IF LEN(S$)>=NUMDIG THEN 7450
7480 LOCATE YFLD,XFLD:PRINT C$;:S$=S$+C$:XFLD=XFLD+1:GOTO 7130
7490 'Backspace
7500 XFLD=XFLD-1:LOCATE YFLD,XFLD:PRINT" ";:IF XFLD<=SXFLD THEN XFLD=SXFLD
7510 LNT=LEN(S$)-1:IF LNT<0 THEN LNT=0
7520 S$=MID$(S$,1,LNT):GOTO 7130
7530 'Number Check routine
7540 NUMCK=0:DECPT=0:EXPNT=0:SIGN=0
7550 IF S$="" THEN 7770 ELSE T$=LEFT$(S$,1)
7560 IF T$="+" OR T$="-" THEN N$=T$:S$=MID$(S$,2)
7570 IF T$="." THEN N$=T$:DECPT=1:S$=MID$(S$,2)
7580 IF S$="" THEN 7790
7590 IF (T$="+" OR T$="-") AND INSTR(".",LEFT$(S$,1))=1 THEN DECPT=1:T$=LEFT$(S$,1):N$=N$+T$:S$=MID$(S$,2)
7600 IF INSTR(DIGIT$,LEFT$(S$,1))=0 THEN 7790
7610 WHILE S$<>"":T$=LEFT$(S$,1):S$=MID$(S$,2)
7620 IF T$="" THEN 7680
7630 IF (T$="+" OR T$="-") AND RIGHT$(N$,1)<>"E" THEN NUMCK=1:GOTO 7680
7640 IF T$="." THEN IF DECPT=1 OR EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:DECPT=1
7650 IF T$="+" OR T$="-" THEN IF SIGN=1 OR EXPNT=0 THEN NUMCK=1 ELSE N$=N$+T$:SIGN=1
7660 IF T$="E" THEN IF EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:EXPNT=1:DECPT=1
7670 IF T$<> "." AND T$<> "+" AND T$<> "-" AND T$<> "E" THEN N$=N$+T$
7680 WEND:IF NUMCK=1 THEN 7790
7690 'Check for overflow
7700 D=0:N%=INSTR(N$,"E"):IF N%=0 THEN 7770
7710 IF N%=1 THEN 7790
7720 S$=LEFT$(N$,N%-1):C$=MID$(N$,N%+1):S=VAL(S$)
7730 IF S=0 THEN D=0 ELSE D=INT(LOG(ABS(S))/LOG(10))
7740 IF C$<> "" THEN D=D+VAL(C$)
7750 IF D<-37 OR D>37 THEN 7790

```

```

7760 'convert
7770 S=VAL(N$):RETURN
7780 'invalid
7790 NUMCK=1:GOSUB 30760:BEEP:LOCATE YFLD,SXFLD,0:PRINT SPC(NUMDIG)
: XFLD=SXFLD:GOTO 7120
7800 'Comment routine
7810 IF C$=CHR$(13) THEN RETURN
7820 IF LEN(S$)>=NUMDIG THEN RETURN ELSE 7480
7830 '
30000 '***** SUBROUTINES *****
30010 '
30020 'CLEAR KBD BUFFER
30030 DEF SEG=0:POKE 1050,PEEK(1052):DEF SEG
30040 RETURN
30050 '
30060 'OPEN DISK FILE FOR INPUT/OUTPUT
30070 'Calling routine specifies filename$, fileenum, fileapp, & file
tpe
30080 'fileapp=0 no append, fileapp=1 append, filetype=0 output =1 i
nput
30090 ON ERROR GOTO 30150:IF FILEAPP=0 THEN 30110 ELSE FILEAPP=0
30100 OPEN FILENAM$ FOR APPEND AS #FILENUM:GOTO 30140
30110 IF FILETPE=1 THEN 30130
30120 OPEN FILENAM$ FOR OUTPUT AS #FILENUM:GOTO 30140
30130 OPEN FILENAM$ FOR INPUT AS #FILENUM
30140 FERR$="OK"
30145 ON ERROR GOTO 0:RETURN
30150 GOSUB 31260:RESUME 30145
30170 '
30180 'FILE DIRECTORY SUBROUTINE
30190 ON ERROR GOTO 30280
30200 COLOR 12,0:CLS:GOSUB 30030:LOCATE 1,18,0:IF SND%>1 THEN BEEP
30210 PRINT"WHICH DRIVE? (A, B, OR AB for both) >";:COLOR 6,0:INPU
T"" ,COM$
30220 PRINT:COLOR 3,0
30230 IF COM$="a:" OR COM$="A:" OR COM$="a" OR COM$="A" THEN PRINT"
A: \":FILES"A:*. *":GOTO 30310
30240 IF COM$="b:" OR COM$="B:" OR COM$="b" OR COM$="B" THEN PRINT"
B: \":FILES"B:*. *":GOTO 30310
30250 IF COM$="ab:" OR COM$="AB:" OR COM$="ab" OR COM$="AB" THEN 30
260 ELSE 30270
30260 PRINT"A: \":FILES"A:*. *":PRINT"B: \":FILES"B:*. *":GOTO 30310
30270 GOSUB 30760:GOTO 30200
30280 GOSUB 31260:GOSUB 30760:COLOR 15,0:PRINT:PRINT SPC(30) FERR$:
COLOR 7,0
30290 PRINT:PRINT SPC(31) RT$:INFLG=1:GOSUB 7120
30300 IF C$="Y" THEN RESUME 30200 ELSE RESUME 30310
30310 ON ERROR GOTO 0
30320 YFLD=24:GOSUB 31680:LOCATE 24,28,0:COLOR 12,0:PRINT"KILL ANY
FILES? (Y/N)";:INFLG=1:GOSUB 7120:IF C$="N" OR ESCK=1 THEN RETURN
30330 GOSUB 31680:LOCATE 24,32,0:PRINT"FILENAME >";:COLOR 6,0

```

```

30340 YFLD=24: XFLD=43: NUMDIG=14: INFLG=4: GOSUB 7110: IF ESCK=1 THEN R
ETURN
30350 FILENAM$=S$: GOSUB 31530: IF FERR$="OK" THEN 30320 ELSE GOSUB 3
1680: GOSUB 30760
30360 COLOR 15,0: LOCATE 24,13,0: PRINT FERR$;: LOCATE 24,47,0: COLOR 7
,0: PRINT RT$;
30370 INFLG=1: GOSUB 7120: IF C$="N" OR ESCK=1 THEN RETURN ELSE COLOR
12,0: GOTO 30330
30380 '
30390 ' PROGRAM WRITE TO ARCHIVE
30400 ' Calling routine provides COM$
30410 FILENAM$=ARCNAM$: FILENUM=2: FILEAPP=1: FILETPE=0
30420 GOSUB 30090: IF FERR$<>"OK" THEN CLOSE #FILENUM: GOTO 30450 'op
en file
30430 ON ERROR GOTO 30490: PRINT #FILENUM,COM$
30440 CLOSE #FILENUM: ON ERROR GOTO 0: RETURN
30450 CLS: COLOR 15,0: LOCATE 10,28,0: PRINT"ERROR ARCHIVING SESSION"
30460 LOCATE 12,28,0: PRINT FERR$: COLOR 7,0: GOSUB 30760
30470 LOCATE 14,30,0: PRINT RT$: INFLG=1: GOSUB 7120
30480 IF C$="Y" THEN 30420 ELSE RETURN
30490 ON ERROR GOTO 0: GOSUB 31260: RESUME 30450
30500 '
30510 ' FILENAME INFORMATION & DIRECTORY ROUTINE
30520 COLOR 3,0: CLS
30530 LOCATE 2,15,0: PRINT"A filename is 1-8 letters/digits. A file
name may"
30540 LOCATE 3,15,0: PRINT"be preceded by a disk drive letter and a
colon,"
30550 LOCATE 4,15,0: PRINT"otherwise the current logged drive is use
d."
30560 IF FILETPE=0 THEN COLOR 15,0: LOCATE 6,8,0: PRINT"WARNING! Usi
ng an existing filename will erase the existing file."
30570 COLOR 3,0: LOCATE 8,24,0: PRINT"Would you like to see your"
30580 COLOR 12,0: LOCATE 9,26,0: PRINT"File Directory? (Y/N)"
30590 INFLG=1: GOSUB 7120: IF C$="Y" THEN GOSUB 30190
30600 RETURN
30610 '
30620 ' INPUT FILENAME ROUTINE
30630 ' calling routine supplies filext$, filenum, filetype, fileapp
30640 ' leaves file open if OK, closes if FO=1
30650 CLS: COLOR 3,0: LOCATE 2,24,0: PRINT"Enter filename including dr
ive."
30660 LOCATE 3,20,0: PRINT"File will be given the extension "FILEXT$
30670 LOCATE 5,30,0: COLOR 12,0: PRINT"FILENAME >"
30680 XFLD=41: YFLD=5: NUMDIG=10: INFLG=4: GOSUB 7110: IF ESCK=1 THEN RE
TURN
30690 FILENAM$=S$+FILEXT$: GOSUB 30090
30700 IF FERR$="OK" THEN RETURN ELSE CLOSE #FILENUM
30710 COLOR 15,0: GOSUB 30760: LOCATE 7,28,0: PRINT FERR$: COLOR 7,0
30720 LOCATE 9,30,0: PRINT RT$: INFLG=1: GOSUB 7120
30730 IF C$="Y" THEN 30650 ELSE RETURN

```

```

30740 '
30750 'FLUB SOUND SUBROUTINE
30760 IF SND%<1 THEN RETURN ELSE SOUND 1000,2: SOUND 1180,3: BEEP: RET
URN
30770 '
30780 'PRESS ANY KEY ROUTINE
30790 'Caller specifies XFLD & YFLD coordinates of message
30800 COLOR 7,0: LOCATE YFLD,XFLD,0: PRINT"Press any key.....";:GOS
UB 30030
30810 IF SND%>1 THEN BEEP
30820 C$=INKEY$: IF C$="" THEN 30820
30830 IF C$=CHR$(27) THEN ESCK=1 ELSE ESCK=0
30840 RETURN
30850 COLOR 7,0: PRINT CHR$(13) CHR$(13) SPC(29)"Press any key.....
.";:GOSUB 30030:GOTO 30810
30860 '
31250 'ERROR CHECKING ROUTINE
31260 IF ERR=64 THEN FERR$="BAD FILE NAME":RETURN
31270 IF ERR=70 THEN FERR$="DISK WRITE PROTECT":RETURN
31280 IF ERR=53 THEN FERR$="FILE NOT FOUND":RETURN
31290 IF ERR=67 THEN FERR$="TOO MANY FILES ON DISK":RETURN
31300 IF ERR=55 THEN FERR$="FILE ALREADY IN USE":RETURN
31310 IF ERR=71 THEN FERR$="DISK NOT READY":RETURN
31320 IF ERR=61 THEN FERR$="DISK FULL":RETURN
31330 IF ERR=72 THEN FERR$="BAD DISK MEDIA":RETURN
31340 FERR$="DISK I/O ERROR":RETURN
31350 '
31510 'KILL FILE ROUTINE
31520 'Caller specifies FILENAM$
31530 ON ERROR GOTO 31550:KILL FILENAM$:FERR$="OK"
31540 ON ERROR GOTO 0:RETURN
31550 GOSUB 31260:RESUME 31540
31570 '
31670 'ERASE ANY LINE ROUTINE
31680 LOCATE YFLD,1,0: PRINT SPC(79);:RETURN
31690 '
31700 'WRITE BLANK TO ARCHIVE
31710 COM$="":GOSUB 30410:RETURN
31720 '
32695 '
32700 GOSUB 30760: SCREEN 0,1: WIDTH 80: CLS: COLOR 15,0: LOCATE 10,28,0
: PRINT COM$
32710 COLOR 7,0: LOCATE 12,33,0: PRINT RT$: INFLG=1: GOSUB 7120: RETURN

```

C.2 ROOT 3/6/85

```

10 COMMON HELP$,HELP$,ARC%,SND%,QW,ARCNAM$:OPTION BASE 1:DEFINT I,J
20 RL$="-----ROOT LOCUS-----"
30 DIM A(13),B(13),C(13),D(13),SG(13),SP0(13,2),SW(13,2)
40 DIM OP(501,25),PX(501),PY(501),KDIM(6),A%(10),B$(10)
42 DEF FNCRD(XR,XI,YR,YI)=(XR*YR+XI*YI)/(YRC2+YIC2)
44 DEF FNCRD(XR,XI,YR,YI)=(XR*YR-XI*YI)/(YRC2+YIC2)
46 DEF FNCRM(XR,XI,YR,YI)=(XR*YR-XI*YI)
48 DEF FNCRD(XR,XI,YR,YI)=(XR*YI+XI*YR)
50 IF QW<1 THEN HELP%=0:HELP$="OFF":ARC%=0:SND%=1
60 MAXDIM=12:MAXPTS=500:DIGIT$="1234567890":RT$="RETRY? (Y/N)"
70 EM=0:IF ARC%=1 THEN GOSUB 31710:COM$=RL$:GOSUB 30410:GOSUB 31710
80 CLS:WIDTH 80:COLOR 14,0:LOCATE 3,31,0:PRINT"ROOT LOCUS"
90 COLOR 10,0:LOCATE 6,23,0:PRINT"<F1>.....ENTER NEW MODEL"
100 LOCATE 7,23,0:PRINT"<F2>.....RECALL OLD MODEL"
110 LOCATE 8,23,0:PRINT"<F3>.....MODIFY CURRENT MODEL"
120 LOCATE 9,23,0:PRINT"<F4>.....STORE MODEL ON DISK"
130 LOCATE 10,23,0:PRINT"<F5>.....ENTER INPUT PARAMETERS"
140 LOCATE 11,23,0:PRINT"<F6>.....CALCULATE DATA"
150 LOCATE 12,23,0:PRINT"<F7>.....GRAPH DATA"
160 LOCATE 13,23,0:PRINT"<F8>.....LIST/SAVE DATA"
170 LOCATE 14,23,0:PRINT"<F9>.....FILE DIRECTORY"
180 LOCATE 15,23,0:PRINT"<ESC>.....EXIT TO MAIN MENU"
190 LOCATE 16,23,0:PRINT"<Alt-F1>....HELP LEVEL"
200 LOCATE 17,23,0:PRINT"<Alt-F2>....SOUND LEVEL"
210 COLOR 14,0:LOCATE 16,46,0:PRINT HELP$:LOCATE 17,46,0:PRINT SND%
220 COLOR 10,0:LOCATE 18,23,0:PRINT"<Alt-F3>....ARCHIVE/COMMENTS"
230 COLOR 14,0:LOCATE 21,24,0:PRINT"PRESS DESIRED FUNCTION KEY"
240 INFLG=2:GOSUB 7120:IF ESCK=1 AND QW=1 THEN CHAIN"DACS"
245 IF ESCK=1 AND QW<>1 THEN 382      'system
250 IF C=59 THEN GOSUB 630:GOTO 80    'new model
260 IF C=60 THEN GOSUB 1100:GOTO 80   'recall model
270 IF C=61 THEN GOSUB 1700:GOTO 80   'modify
280 IF C=62 THEN GOSUB 940:GOTO 80    'store model
290 IF C=63 THEN GOSUB 1340:GOTO 80   'input params
300 IF C=64 THEN GOSUB 1580:GOTO 80   'run
310 IF C=65 THEN GOSUB 2820:GOTO 80   'graph
320 IF C=66 THEN GOSUB 3590:GOTO 80   'list
330 IF C=67 THEN GOSUB 30190:GOTO 80  'files
340 IF C=104 THEN GOSUB 590:GOTO 210  'help
350 IF C=105 THEN GOSUB 550:GOTO 210  'help
360 IF C=106 THEN GOSUB 390:GOTO 80   'archive
370 GOSUB 30760:GOTO 240
380 '
381 'SYSTEM
382 CLEAR:WIDTH 80:SCREEN 0,0:COLOR 7,0:CLOSE:END
383 '
385 'ARCHIVE
390 IF ARC%=1 THEN GOSUB 420:RETURN
395 GOSUB 420:IF ARC%=0 THEN RETURN ELSE GOSUB 31710:COM$=RL$:GOSUB

```

```

30410:GOSUB 31710:RETURN
400 '
410 ' ARCHIVE SESSION / ADD COMMENTS
420 COLOR 12,0:CLS:IF ARC%=-1 THEN 470 ELSE LOCATE 2,27,0
430 PRINT"ARCHIVE SESSION? (Y/N)":INFLG=1:GOSUB 7120:IF ESCK=1 OR
C$="N" THEN RETURN
440 FILETPE=0:IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
450 FILEXT$=".ARC":FILENUM=2:GOSUB 30650:CLOSE #FILENUM
460 IF ESCK=1 OR FERR$<>"OK" THEN RETURN ELSE ARCNAM$=FILENAM$:ARC%
=1:CLS
470 COLOR 12,0:LOCATE 2,28,0:PRINT"ENTER COMMENTS? (Y/N)":INFLG=1:
GOSUB 7120
480 IF C$="N" OR ESCK=1 THEN RETURN ELSE NUMDIG=78:INFLG=5:XFLD=1
490 CLS:LOCATE 2,19,0:YFLD=4
500 PRINT"ENTER COMMENTS BELOW....PRESS ESC TO QUIT":PRINT
510 GOSUB 7110:IF ESCK=1 THEN GOSUB 31710:RETURN
520 COM$=S$:GOSUB 30410:IF FERR$<>"OK" THEN RETURN ELSE YFLD=YFLD+1
:XFLD=1:IF YFLD>24 THEN 490 ELSE 510
530 '
540 ' SOUND LEVEL
550 SND%=SND%+1:IF SND%>2 THEN SND%=0
560 RETURN
570 '
580 ' SET HELP LEVEL
590 IF HELP%=1 THEN HELP%=0:HELP$="OFF":RETURN
600 HELP%=1:HELP$="ON ":RETURN
610 '
620 ' ENTER NEW MODEL
630 IF EM=1 THEN COM$="ERASE EXISTING MODEL? (Y/N)":GOSUB 32340:IF
C$="N" OR ESCK=1 THEN RETURN
640 CLS:IF HELP%=0 THEN 755 ELSE COLOR 3,0
650 LOCATE 2,12,0:PRINT"The Root Locus program calculates the zeros
of the equation"
660 LOCATE 4,34,0:COLOR 15,0:PRINT"1 + G(s)":COLOR 3,0
670 LOCATE 6,12,0:PRINT"as a function of K. G(s) is assumed ration
al and"
680 LOCATE 8,29,0:COLOR 15,0:PRINT"G(s) = KN(s)/D(s)":COLOR 3,0
690 LOCATE 10,12,0:PRINT"You will be prompted for the polynomials N
(s) & D(s). "
700 LOCATE 11,12,0:PRINT"The coefficient of the highest order term
is assumed unity."
710 LOCATE 12,12,0:PRINT"Enter the coefficients in ascending order.
Example:"
720 LOCATE 14,20,0:COLOR 15,0:PRINT"G(s) = (s + 3)/(s**3 + 2s**2 +
9s + 3)"
730 LOCATE 16,30,0:PRINT"N(s) = 3":LOCATE 17,30,0:PRINT"D(s) = 3
9 2":COLOR 3,0
740 LOCATE 19,12,0:PRINT"The coefficient of the highest order term
is not entered."
750 GOSUB 30850:IF ESCK=1 THEN 910 ELSE CLS
755 GOSUB 760:IF ESCK=1 THEN 910 ELSE GOSUB 850:IF ESCK=1 THEN 910

```

```

756 EM=1: IF ARC%=1 THEN GOSUB 31710: COM$="-ENTER NEW MODEL-": GOSUB
30410
757 RETURN
760 COLOR 12,0: LOCATE 10,30,0: PRINT"N(s) DIMENSION >"SPC(20)
770 XFLD=47: YFLD=10: INFLG=3: NUMDIG=12: GOSUB 7110: IF ESCK=1 THEN RET
URN
780 S=INT(S): IF S<=MAXDIM-1 AND S>=0 THEN KDIM(1)=S: GOTO 800
790 GOSUB 30760: COLOR 15,0: LOCATE 12,18,0: PRINT"ERROR! Value must
lie in the range 0 to"MAXDIM-1: GOTO 760
800 COLOR 12,0: LOCATE 14,30,0: PRINT"D(s) DIMENSION >"SPC(20)
810 XFLD=47: YFLD=14: INFLG=3: NUMDIG=12: GOSUB 7110: IF ESCK=1 THEN RET
URN
820 S=INT(S): IF S<=MAXDIM AND S>0 AND S>KDIM(1) THEN KDIM(2)=S: RETU
RN
830 COLOR 15,0: LOCATE 16,18,0: PRINT"ERROR! Value must lie in the r
ange 1 to"MAXDIM: GOSUB 30760
840 LOCATE 17,18,0: PRINT"and be greater than the dimension of N(s).
": GOTO 800
850 CLS: NUMDIG=10: INFLG=3: P=0: IF HELP%=0 THEN 870 ELSE COLOR 3,0
860 LOCATE 1,3,0: PRINT"ENTER POLYNOMIALS IN ASCENDING ORDER PRESSIN
G <ENTER> AFTER EACH ELEMENT.";
870 A(KDIM(1)+1)=1: B(KDIM(2)+1)=1: IF KDIM(1)=0 THEN 880
875 MAT=1: COM$="N(s) POLYNOMIAL": XFLD=1: YFLD=3: ROW%=1: COL%=KDIM(1):
GOSUB 30890: IF ESCK=1 THEN RETURN
880 MAT=2: COM$="D(S) POLYNOMIAL": XFLD=1: YFLD=9: ROW%=1: COL%=KDIM(2):
GOSUB 30890: RETURN
910 COM$="MODEL NOT ENTERED!!": GOSUB 32310: EM=0: RETURN
920 '
930 ' SAVE MODEL ON DISK
940 IF EM=0 THEN COM$="NO MODEL TO SAVE!!": GOSUB 32310: RETURN
950 FILETPE=0: IF HELP%=1 THEN GOSUB 30510 ELSE CLS: GOSUB 30580
960 FILEXT$=".TMD": FILENUM=1: GOSUB 30650
970 IF ESCK=0 AND FERR$="OK" THEN 990
980 COM$="MODEL NOT SAVED!!": GOSUB 32310: RETURN
990 SFILE$=FILENAM$: CLS: COLOR 3,0: LOCATE 10,30,0: PRINT"SAVING MODEL
....."
1000 IF ARC%=1 THEN GOSUB 31710: COM$="-STORE MODEL ON DISK-": GOSUB
30410: FILENUM=1
1010 MAT=0: ROW%=2: COL%=1: GOSUB 31380: IF FERR$<>"OK" THEN 1050
1020 MAT=1: ROW%=1: COL%=KDIM(1)+1: GOSUB 31380: IF FERR$<>"OK" THEN 10
50
1030 MAT=2: ROW%=1: COL%=KDIM(2)+1: GOSUB 31380: IF FERR$<>"OK" THEN 10
50
1035 CLOSE#FILENUM: IF ARC%=1 THEN COM$="Model saved under filename"
+SFILE$: GOSUB 30410
1040 RETURN
1050 COM$=FERR$: GOSUB 32700: FILENAM$=SFILE$: GOSUB 31530: IF C$="Y" A
ND ESCK=0 THEN 950
1060 IF ARC%=1 THEN COM$=FERR$: GOSUB 30410
1070 GOTO 980
1080 '

```

```

1090 'RECALL MODEL ROUTINE
1100 IF EM=1 THEN COM$="ERASE EXISTING MODEL? (Y/N)":GOSUB 32340: I
F C$="N" OR ESCK=1 THEN RETURN
1110 FILETPE=1: IF HELP%=1 THEN GOSUB 30520 ELSE CLS:GOSUB 30580
1120 FILEXT$=".TMD": FILENUM=1:GOSUB 30650
1130 IF ESCK=0 AND FERR$="OK" THEN SFILE$=FILENAM$:GOTO 1150
1140 COM$="MODEL NOT RECALLED!!":GOSUB 32310: RETURN
1150 CLS: COLOR 3,0: LOCATE 10,30,0: PRINT"RECALLING MODEL....."
1160 IF ARC%=1 THEN GOSUB 31710: COM$="-RECALL MODEL-":GOSUB 30410: F
ILENUM=1
1170 ROW%=2: COL%=1: MAT=0: GOSUB 31110: IF FERR$<>"OK" THEN 1290
1180 ROW%=1: COL%=KDIM(1)+1: MAT=1: GOSUB 31110: IF FERR$<>"OK" THEN 12
90
1190 ROW%=1: COL%=KDIM(2)+1: MAT=2: GOSUB 31110: IF FERR$<>"OK" THEN 12
90 ELSE EM=1
1200 CLOSE #FILENUM: COLOR 7,0: LOCATE 14,30,0: PRINT"SEE MODEL? (Y/N
)"
1210 INFLG=1: GOSUB 7120: IF C$="Y" AND ESCK=0 THEN 1240
1220 IF ARC%=1 THEN COM$="Model recalled from file"+SFILE$:GOSUB 30
410
1230 RETURN
1240 CLS: COLOR 12,0: LOCATE 10,31,0: PRINT"N(s) DIMENSION >": LOCATE 1
2,31,0: PRINT"D(s) DIMENSION >"
1250 COLOR 6,0: LOCATE 10,48,0: PRINT KDIM(1): LOCATE 12,48,0: PRINT KD
IM(2)
1260 GOSUB 30850: IF ESCK=1 THEN 1220 ELSE P=1: NUMDIG=10: CLS
1270 YFLD=3: XFLD=1: MAT=1: COM$="N(s) POLYNOMIAL": ROW%=1: COL%=KDIM(1)
+1: GOSUB 30890
1280 YFLD=9: XLFD=1: MAT=2: COM$="D(s) POLYNOMIAL": ROW%=1: COL%=KDIM(2)
+1: GOSUB 30890: GOSUB 30850: GOTO 1220
1290 CLOSE #FILENUM: COM$=FERR$: GOSUB 32700: IF C$="Y" AND ESCK=0 THE
N 1110 ELSE EM=0
1300 IF ARC%=1 THEN COM$=FERR$+"for file"+SFILE$:GOSUB 30410
1310 GOTO 1140
1320 '
1330 ' INPUT PARAMTERS
1340 IF EM=0 THEN COM$="ENTER MODEL FIRST!!":GOSUB 32310: RETURN
1350 CLS: IP=0: IF HELP%=0 THEN 1430 ELSE COLOR 3,0
1360 LOCATE 2,12,0: PRINT"You will be asked to enter the minimum gai
n, KMIN, the"
1370 LOCATE 3,12,0: PRINT"maximum gain, KMAX, and the gain interval,
KINT. The"
1380 LOCATE 4,12,0: PRINT"number of points to be calculated equals":
COLOR 15,0
1390 LOCATE 6,28,0: PRINT"(KMAX - KMIN) / KINT.":COLOR 3,0
1400 LOCATE 8,22,0: PRINT"This number may not exceed"MAXPTS
1420 GOSUB 30850: IF ESCK=1 THEN 1550 ELSE CLS
1430 COLOR 12,0: LOCATE 6,31,0: PRINT"MINIMUM GAIN >"SPC(20)
1440 XFLD=48: YFLD=6: NUMDIG=12: INFLG=3: GOSUB 7110: IF ESCK=1 THEN 155
0 ELSE KDIM(3)=S
1460 COLOR 12,0: LOCATE 10,31,0: PRINT"MAXIMUM GAIN >"SPC(20)

```

```

1470 XFLD=48: YFLD=10: NUMDIG=12: INFLG=3: GOSUB 7110: IF ESCK=1 THEN 15
50
1480 IF S>KDIM(3) THEN KDIM(4)=S: GOTO 1490
1482 GOSUB 30760: COLOR 15,0: LOCATE 12,15,0
1485 PRINT"ERROR! Value must be greater than minimum gain.": GOTO 1
460
1490 COLOR 12,0: LOCATE 14,31,0: PRINT"GAIN INTERVAL >"SPC(20)
1500 XFLD=48: YFLD=14: NUMDIG=12: INFLG=3: GOSUB 7110: IF ESCK=1 THEN 15
50 ELSE KDIM(5)=S
1505 IF KDIM(5)<=0 THEN 1520 ELSE PN=(KDIM(4)-KDIM(3))/KDIM(5)
1510 IF PN>=1 AND PN<=MAXPTS THEN 1530
1520 GOSUB 30760: COLOR 15,0: LOCATE 16,10,0
1525 PRINT"ERROR! (maximum-minimum)/interval must lie between 1 an
d"MAXPTS: GOTO 1490
1530 IF ARC%=1 THEN GOSUB 31710: COM$="-INPUT PARAMETERS=": GOSUB 304
10
1540 IP=1: RETURN
1550 IP=0: COM$="INPUTS NOT ENTERED!": GOSUB 32310: RETURN
1560 '
1570 'CALCULATE DATA
1580 IF EM=0 THEN COM$="ENTER MODEL FIRST!": GOSUB 32310: RETURN
1590 IF IP=0 THEN COM$="ENTER INPUTS FIRST!": GOSUB 32310: RETURN
1600 RS=1: CLS: LOCATE 11,29,0: COLOR 3,0: PRINT"CALCULATING DATA.....
"
1610 IF ARC%=1 THEN GOSUB 31710: COM$="-CALCULATE DATA-": GOSUB 30410
1620 FOR I=KDIM(1)+2 TO MAXDIM+1: A(I)=0: NEXT I
1630 FOR I=KDIM(2)+2 TO MAXDIM+1: B(I)=0: NEXT I
1640 K=KDIM(3): NX=KDIM(2): ITER=1
1650 FOR I=1 TO NX+1: C(I)=A(I)*K+B(I): NEXT I: N=NX: GOSUB 32790: OP(IT
ER,1)=K
1660 FOR I=2 TO NX*2+1 STEP 2: OP(ITER,I)=SP0(I/2,1): OP(ITER,I+1)=SP
0(I/2,2): NEXT I
1670 ITER=ITER+1: K=K+KDIM(5): IF K-KDIM(4)<=0 THEN 1650 ELSE RETURN
1680 '
1690 'MODIFY CURRENT MODEL
1700 IF EM=0 THEN COM$="NO MODEL TO MODIFY!": GOSUB 32310: RETURN
1710 IF ARC%=1 THEN GOSUB 31710: COM$="-MODIFY CURRENT MODEL-": GOSUB
30410
1720 CLS: IF HELP%=0 THEN 1740 ELSE LOCATE 1,13,0: COLOR 3,0
1730 PRINT"ENTER LETTER OF POLYNOMIAL TO CHANGE, ESC TO QUIT"
1740 COLOR 12,0: LOCATE 3,30,0: PRINT"(N, D, OR ESC) >"SPC(5): INFLG=5
: NUMDIG=1: YFLD=3: XFLD=49
1750 GOSUB 7110: IF ESCK=1 THEN RETURN
1770 IF S$="N" OR S$="n" THEN COM$="N(s) POLYNOMIAL": MAT=1: ROW%=1: C
OL%=KDIM(1): IF COL%=0 THEN 1790 ELSE GOSUB 32440: GOTO 1720
1780 IF S$="D" OR S$="d" THEN COM$="D(S) POLYNOMIAL": MAT=2: ROW%=1: C
OL%=KDIM(2): GOSUB 32440: GOTO 1720
1790 GOSUB 30760: GOTO 1740
1800 '
2800 '
2810 'GRAPHING ROUTINE

```

```

2820 RG=0: IF ARC%=1 THEN GOSUB 31710: COM$="-GRAPH DATA-": GOSUB 3041
0
2830 SCREEN 0,1: CLS: COLOR 14,0: LOCATE 3,36,0: PRINT"OPTIONS": COLOR 1
0,0
2840 LOCATE 5,24,0: PRINT"<Ctrl-F1>....GRAPH CURRENT DATA"
2850 LOCATE 6,24,0: PRINT"<Ctrl-F2>....RE-DISPLAY SCREEN FILE"
2860 LOCATE 7,24,0: PRINT"<ESC>.....END"
2870 COLOR 14,0: LOCATE 12,29,0: PRINT"WHILE GRAPH DISPLAYED": COLOR 1
0,0
2880 LOCATE 14,24,0: PRINT"<ENTER>.....CURSOR COORDINATES"
2900 LOCATE 15,24,0: PRINT"<Ctrl-F4>....SAVE GRAPH ON DISK"
2910 LOCATE 16,24,0: PRINT"<Ctrl-F5>....ZOOM"
2920 LOCATE 17,24,0: PRINT"<ESC>.....END": COLOR 14,0
2930 LOCATE 19,29,0: PRINT"(EXCEPT ON RE-DISPLAY)"
2940 INFLG=2: GOSUB 7120: IF ESCK=1 THEN RETURN
2950 IF C=94 THEN GOSUB 3000: GOTO 2830
2960 IF C=95 THEN GOSUB 3400: GOTO 2830
2970 GOSUB 30760: GOTO 2940
2980 '
2990 'GRAPH CURRENT DATA
3000 IF RS=0 THEN COM$="NO DATA TO GRAPH!": GOSUB 32310: RETURN
3010 CLS: IF ARC%=1 THEN COM$="Graphed current data": GOSUB 30410
3020 COLOR 12,0: LOCATE 10,28,0: PRINT"GRIDS ON GRAPH? (Y/N)"
3030 IF HELP%=1 THEN LOCATE 11,31,0: PRINT"(Default = No)"
3035 INFLG=1: GOSUB 7110: IF C$="N" THEN PGD=0 ELSE PGD=1
3040 PMINX=0: PMAXX=0: PMINY=0: PMAXY=0: PFM=1
3050 FOR I=1 TO PN: FOR J=2 TO KDIM(2)*2+1 STEP 2
3060 IF OP(I,J)<PMINX THEN PMINX=OP(I,J)
3070 IF OP(I,J)>PMAXX THEN PMAXX=OP(I,J)
3080 IF OP(I,J+1)<PMINY THEN PMINY=OP(I,J+1)
3090 IF OP(I,J+1)>PMAXY THEN PMAXY=OP(I,J+1)
3100 NEXT J,I: PFLG=1: GOSUB 4620: IF P=0 THEN RETURN
3110 PFLG=2: FOR I=2 TO KDIM(2)*2+1 STEP 2: FOR J=1 TO PN
3120 PX(J)=OP(J,I): PY(J)=OP(J,I+1): NEXT J: GOSUB 4620: NEXT I
3130 GOSUB 6020: IF ZOOM=0 THEN RETURN ELSE PFLG=1: GOSUB 4620
3140 IF P=0 THEN RETURN ELSE GOTO 3110
3380 '
3390 'RE-GRAPH SCREEN FILE
3400 CLS: FILETPE=1: IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
3410 FILEXT$=".GPH": FILENUM=1: GOSUB 30650
3420 IF ESCK=1 OR FERR$<>"OK" THEN RETURN ELSE CLOSE #FILENUM: RG=1
3430 CLS: SCREEN 2: DEF SEG=&HB800
3440 ON ERROR GOTO 3510: BLOAD FILENAM$,0: DEF SEG: ON ERROR GOTO 0
3450 IF ARC%=1 THEN COM$="Re-graph from screen file "+FILENAM$: GOSUB
B 30410
3470 INFLG=2: GOSUB 7120: IF ESCK=1 THEN RETURN
3500 GOSUB 30760: GOTO 3470
3510 ON ERROR GOTO 0: CLS: SCREEN 0,1: COLOR 15,0: GOSUB 30760: LOCATE 1
0,30,0
3520 PRINT"ERROR GRAPHING FILE": COLOR 7,0: LOCATE 12,33,0: PRINT RT$:
INFLG=1

```

```

3530 GOSUB 7120: IF C$="Y" AND ESCK=0 THEN RESUME 3400 ELSE RESUME 3
540
3540 IF ARC%=1 THEN COM$="ERROR REGRAPHING FILE "+FILENAM$: GOSUB 30
410
3560 RETURN
3570 '
3580 ' SAVE/LIST DATA
3590 IF ARC%=1 THEN GOSUB 31710: COM$="-SAVE/LIST DATA-": GOSUB 30410
3600 CLS: COLOR 14,0: LOCATE 4,36,0: PRINT "OPTIONS": COLOR 10,0
3610 LOCATE 6,25,0: PRINT "<Ctrl-F6>....LIST CURRENT DATA"
3620 LOCATE 7,25,0: PRINT "<Ctrl-F7>....SAVE DATA ON DISK"
3630 LOCATE 8,25,0: PRINT "<Ctrl-F8>....RECALL DATA FILE"
3640 LOCATE 9,25,0: PRINT "<ESC>.....END"
3650 INFLG=2: GOSUB 7120: IF ESCK=1 THEN RETURN
3660 IF C=99 THEN GOSUB 3720: GOTO 3600
3670 IF C=100 THEN GOSUB 3890: GOTO 3600
3680 IF C=101 THEN GOSUB 4070: GOTO 3600
3690 GOSUB 30760: GOTO 3650
3700 '
3710 ' LIST CURRENT DATA
3720 IF RS=0 THEN COM$="NO DATA TO LIST!": GOSUB 32310: RETURN
3730 IF ARC%=1 THEN COM$="Listed current data": GOSUB 30410
3740 FOR I=1 TO PN: CLS: COLOR 12,0: LOCATE 1,35,0: PRINT "GAIN = "
3750 LOCATE 3,31,0: PRINT "REAL      IMAGINARY": COLOR 6,0: LOCATE 1,42
,0: PRINT OP(I,1)
3760 YFLD=5: XFLD=28: FOR J=2 TO KDIM(2)*2+1 STEP 2: LOCATE YFLD,XFLD,
0: PRINT OP(I,J)
3770 LOCATE YFLD,XFLD+14,0: PRINT OP(I,J+1): YFLD=YFLD+1: NEXT J: GOSUB
30850
3780 IF ESCK=1 THEN RETURN ELSE NEXT I: RETURN
3870 '
3880 ' SAVE DATA ON DISK
3890 IF RS=0 THEN COM$="NO DATA TO SAVE!": GOSUB 32310: RETURN
3910 CLS: FILETPE=0: IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
3920 FILEXT$=".RDT": FILENUM=1: GOSUB 30650: KDIM(6)=PN
3930 IF ESCK=0 AND FERR$="OK" THEN 3960
3940 COM$="DATA NOT SAVED!": GOSUB 32310: RETURN
3960 SFILE$=FILENAM$: CLS: COLOR 3,0: LOCATE 11,31,0: PRINT "SAVING DATA
....."
3970 MAT=0: ROW%=6: COL%=1: GOSUB 31380: IF FERR$<>"OK" THEN 4010
3980 MAT=6: ROW%=PN: COL%=KDIM(2)*2+1: GOSUB 31380: IF FERR$<>"OK" THEN
4010 ELSE CLOSE #FILENUM
3990 IF ARC%=1 THEN COM$="Data saved in file "+SFILE$: GOSUB 30410
4000 RETURN
4010 CLOSE #FILENUM: COM$=FERR$: GOSUB 32700: IF C$="Y" AND ESCK=0 THE
N 3910
4030 IF ARC%=1 THEN COM$=FERR$+" while saving data": GOSUB 30410
4040 FILENAM$=SFILE$: GOSUB 31530: GOTO 3940
4050 '
4060 ' RECALL DATA FILE
4070 IF RS=1 THEN COM$="ERASE EXISTING DATA? (Y/N)": GOSUB 32340: IF

```

```

C$='N' OR ESCK=1 THEN RETURN
4100 CLS:FILETPE=1: IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
4110 FILEXT$='.RDT': FILENUM=1: GOSUB 30650
4120 IF ESCK=0 AND FERR$='OK' THEN SFILE$=FILENAM$: GOTO 4160
4130 COM$='DATA NOT RECALLED!!': GOSUB 32310: RETURN
4160 CLS: COLOR 3,0: LOCATE 10,29,0: PRINT"RECALLING DATA....."
4165 TEMP1=KDIM(3): TEMP2=KDIM(4): TEMP3=KDIM(5)
4170 ROW%=6: COL%=1: MAT=0: GOSUB 31110: IF FERR$<>"OK" THEN 4210 ELSE
PN=KDIM(6)
4175 KDIM(3)=TEMP1: KDIM(4)=TEMP2: KDIM(5)=TEMP3
4180 ROW%=PN: COL%=KDIM(2)*2+1: MAT=6: GOSUB 31110: IF FERR$<>"OK" THEN
4210
4190 RS=1: CLOSE #FILENUM: IF ARC%=1 THEN COM$='Recalled data from fi
le "+SFILE$: GOSUB 30410
4200 RETURN
4210 CLOSE #FILENUM: COM$=FERR$: GOSUB 32700: IF C$='Y' AND ESCK=0 THE
N 4100 ELSE RS=0: GOTO 4130
4570 '
4580 ' PLOTTING SUBROUTINE
4590 ' PFLG=0 -axes & plot : PFLG=1 -axes only : PLFG=2 -plot only
4600 ' PGD=1 grids on plot : PGD=0 no grids : PFM=1 don't find min/m
ax
4610 ' Plots PX() and PY() arrays where PN = number of xy pairs
4620 IF PFLG>1 THEN GOSUB 5490: RETURN
4630 GOSUB 4720
4640 IF PFM=1 THEN GOSUB 5170: IF P=0 THEN RETURN ELSE GOSUB 4680: GO
SUB 5240: GOTO 4660
4650 GOSUB 4820: GOSUB 5170: IF P=0 THEN RETURN ELSE GOSUB 4680: GOSUB
5240
4660 IF PFLG=1 THEN RETURN ELSE GOSUB 5500
4670 RETURN
4680 ' TITLE
4690 LOCATE 1,7
4700 PRINT"SCALE X-AXIS BY";STR$(PSCLX);", Y-AXIS BY";PSCLY;
4710 RETURN
4720 ' INITIALIZATION
4730 DEF FNPSC(PS,PXS,PNS,PXO,PNO)=INT(PXO-(PXO-PNO)/(PXS-PNS)*(PXS
-PS)+.5)
4750 PNINX=6.5: PNINY=5.5: SCREEN 2: CLS
4760 PLNL=50: PLNR=589: PLNB=173: PLNT=12: PLOCX1=8
4770 PLOCY1=INT(PLNT/8+.5): PLOCY2=PLOCY1+1
4780 PLOCY3=INT(PLNB/8+.625): PLOCY4=PLOCY3+1
4790 PLOCY5=PLOCY4+1: PLOCX2=INT(PLNL/8-4)
4800 PLOCX3=PLOCX2+1: PLOCX4=PLOCX3+INT((PLNR-PLNL)/8+.625)-1
4810 RETURN
4820 ' FIND MINIMUM & MAXIMUM OF ARRAYS
4830 PMINX=PX(1): PMAXX=PMINX: PMINY=PY(1): PMAXY=PMINY
4840 FOR PII%=2 TO PN
4850 IF PMINX>PX(PII%) THEN PMINX=PX(PII%)
4860 IF PMAXX<PX(PII%) THEN PMAXX=PX(PII%)
4870 IF PMINY>PY(PII%) THEN PMINY=PY(PII%)

```

```

4880 IF PMAXY<PY(PII%) THEN PMAXY=PY(PII%)
4890 NEXT PII%
4900 RETURN
4910 'FIND LINEAR SCALE INTERVALS
4920 IF PMIN=PMAX THEN PMIN=PMIN-1: PMAX=PMAX+1
4950 P=1: PEPS=.025: PA=ABS(PMIN)
4960 IF ABS(PMIN)<ABS(PMAX) THEN PA=ABS(PMAX)
4970 PSCL=10*INT(LOG(PA)/LOG(10))
4980 PMINA=PMIN/PSCL: PMAXA=PMAX/PSCL
4990 PD=(PMAXA-PMINA)/PNIN: PJ=PD*PEPS: PE=INT(LOG(PD)/LOG(10))
5000 PF=PD/10*PE: PV=10
5010 IF PF<SQR(50) THEN PV=5
5020 IF PF<SQR(10) THEN PV=2
5030 IF PF<SQR(2) THEN PV=1
5040 PWDT=PV*10*PE: PG=INT(PMINA/PWDT)
5050 IF ABS(PG+1-PMINA/PWDT)<PJ THEN PG=PG+1
5060 PMINP=PWDT*PG: PH=INT(PMAXA/PWDT)+1
5070 IF ABS(PMAXA/PWDT+1-PH)<PJ THEN PH=PH-1
5080 PMAXP=PWDT*PH: PNINP=PH-PG
5090 IF ABS(PMAXP)>=10 OR ABS(PMINP)>=10 THEN PSCL=PSCL*10: GOTO 498
0
5100 RETURN
5110 'LINEAR TRANSFORMATION FROM SUBJECT TO OBJECT SPACE
5120 PX1=FNPSC(PX(PII%)/PSCLX, PMAXPX, PMINPX, PLNR, PLNL)
5130 PY1=FNPSC(PY(PII%)/PSCLY, PMAXPY, PMINPY, PLNT, PLNB)
5160 RETURN
5170 'FIND X & Y LINEAR SCALE INTERVALS
5180 'X SCALE
5190 PMIN=PMINX: PMAX=PMAXX: PNIN=PNINX: GOSUB 4910: IF P=0 THEN RETURN
5200 PSCLX=PSCL: PMINPX=PMINP: PMAXPX=PMAXP: PNINPX=PNINP: PWDTX=PWDT
5210 'Y SCALE
5220 PMIN=PMINY: PMAX=PMAXY: PNIN=PNINY: GOSUB 4910: IF P=0 THEN RETURN
5230 PSCLY=PSCL: PMINPY=PMINP: PMAXPY=PMAXP: PNINPY=PNINP: PWDY=PWDT: R
ETURN
5240 'DRAW LINEAR GRIDS & LABELS
5250 PLOCWY=(PLOCY3-PLOCY2+1)/PNINPY: PLNWY=(PLNB-PLNT)/PNINPY: X1=PL
NL-49: X2=PLNL-4
5260 'Y GRID & LABELS
5270 FOR PII=0 TO PNINPY: PLNY1=PLNB-PII*PLNWY
5280 Y1=PLNY1-5: Y2=PLNY1+5: S$=STR$(PMINPY+PII*PWDY): GOSUB 6850
5290 IF PGD=1 THEN 5320
5300 IF PII=0 OR PII=PNINPY THEN 5320
5310 LINE(PLNL-4, PLNY1)-(PLNL, PLNY1): LINE(PLNR, PLNY1)-(PLNR+4, PLNY1
): GOTO 5330
5320 LINE(PLNL, PLNY1)-(PLNR, PLNY1)
5330 NEXT PII
5340 PLOCWX=(PLOCX4-PLOCX3+1)/PNINPX: PLNWX=(PLNR-PLNL)/PNINPX
5350 'X GRID & LABELS
5360 FOR PII=0 TO PNINPX
5370 IF PII/2=INT(PII/2) THEN PLOC=PLOCY5 ELSE PLOC=PLOCY4
5380 PLNX1=PLNL+PII*PLNWX: Y1=PLOC*8-8: Y2=Y1+8: X1=PLNX1-30: X2=PLNX1+

```

25

```

5390 S$=STR$(PMINPX+PII*PMDTX):GOSUB 6850:IF PGD=1 THEN 5420
5400 IF PII=0 OR PII=PNINPX THEN 5420
5410 LINE(PLNX1,PLNB)-(PLNX1,PLNB+2):LINE(FLNX1,PLNT)-(PLNX1,PLNT-2
):GOTO 5430
5420 LINE(PLNX1,PLNB)-(PLNX1,PLNT)
5430 NEXT PII
5440 RETURN
5490 'CLIPPING ROUTINE TO CHECK FOR POINTS OUTSIDE GRID
5500 FOR PII%=1 TO PN:GOSUB 5120
5510 IF PX1<PLNL OR PX1>PLNR OR PY1>PLNB OR PY1<PLNT THEN 5530
5520 PSET(PX1,PY1):PSET(PX1-1,PY1):PSET(PX1+1,PY1):PSET(PX1,PY1+1):
PSET(PX1,PY1-1)
5530 NEXT PII%:RETURN
5990 '
6000 'CURSOR SUBROUTINE
6010 'Moves cursor around screen keeping pixel coordinates
6020 DEF SEG=0:POKE 1047,64:DEF SEG:IF SND%>1 THEN BEEP
6030 CRSRX=PLNL:CRSRY=PLNB:ZOOM=0:GOSUB 6830
6040 IF CRSRX<PLNL THEN CRSRX=PLNL
6050 IF CRSRX>PLNR THEN CRSRX=PLNR
6060 IF CRSRY<PLNT THEN CRSRY=PLNT
6070 IF CRSRY>PLNB THEN CRSRY=PLNB
6080 GOSUB 6300:GOSUB 30030 'Draw csr & clear KBD buf
6090 C$=INKEY$:IF C$="" THEN 6090
6100 IF C$=CHR$(27) THEN ZOOM=0:RETURN
6110 IF C$=CHR$(13) THEN GOSUB 6470 'Wants coordinates
6120 IF LEN(C$)=2 THEN C$=RIGHT$(C$,1) ELSE 6230
6150 IF C$=CHR$(98) THEN GOSUB 6550:IF ESCK<>1 THEN ZOOM=1:RETURN
6160 IF C$=CHR$(97) THEN GOSUB 6670:GOSUB 6830
6170 GOSUB 6380 'Erase CRSR
6180 IF C$=CHR$(72) THEN CRSRY=CRSRY-1:GOTO 6040
6190 IF C$=CHR$(80) THEN CRSRY=CRSRY+1:GOTO 6040
6200 IF C$=CHR$(77) THEN CRSRX=CRSRX+1:GOTO 6040
6210 IF C$=CHR$(75) THEN CRSRX=CRSRX-1:GOTO 6040
6220 DEF SEG=0:POKE 1047,64:DEF SEG:GOTO 6080
6230 GOSUB 6380 'Erase CRSR
6240 IF C$=CHR$(56) THEN CRSRY=CRSRY-5:GOTO 6040
6250 IF C$=CHR$(50) THEN CRSRY=CRSRY+5:GOTO 6040
6260 IF C$=CHR$(54) THEN CRSRX=CRSRX+10:GOTO 6040
6270 IF C$=CHR$(52) THEN CRSRX=CRSRX-10:GOTO 6040
6280 DEF SEG=0:POKE 1047,64:DEF SEG:GOTO 6080
6290 'DRAW CURSOR
6300 CRSRPX=CRSRX-6:CRSRPY=CRSRY-6
6310 FOR CII%=1 TO 11
6320 CRSRPX=CRSRPX+1:CRSRPY=CRSRPY+1
6330 SW(CII%,1)=POINT(CRSRPX,CRSRY):SW(CII%,2)=POINT(CRSRX,CRSRPY)
6340 PSET(CRSRPX,CRSRY):PSET(CRSRX,CRSRPY)
6350 NEXT CII%
6360 RETURN
6370 'ERASE CURSOR

```

```

6380 CRSREX=CRSRX-6: CRSREY=CRSRY-6
6390 FOR CII%=1 TO 11
6400 CRSREX=CRSREX+1: CRSREY=CRSREY+1
6410 PRESET(CRSREX,CRSRY): PRESET(CRSRX,CRSREY)
6420 IF SW(CII%,1)=1 THEN PSET(CRSREX,CRSRY)
6430 IF SW(CII%,2)=1 THEN PSET(CRSRX,CRSREY)
6440 NEXT CII%
6450 RETURN
6460 ' LINEAR COORDINATE FINDER
6470 PBYP=PMINPY: PTY=PMINPY+PNINPY*PBDTY
6480 PLX=PMINPX: PRX=PMINPX+PNINPX*PBDTX
6490 PXCOD=(PRX-(PLNR-CRSRX)*(PRX-PLX)/(PLNR-PLNL))*PSCLX
6500 PYCOD=(PTY-(PLNB-CRSRY)*(PTY-PTY)/(PLNB-PLNT))*PSCLY
6510 LOCATE 1,47: PRINT USING "X= ##.###ccc"; PXCOD
6520 LOCATE 1,64: PRINT USING "Y= ##.###ccc"; PYCOD: RETURN
6530 '
6540 ' ZOOM ROUTINE
6550 LOCATE 1,47,0: PRINT "X1=" SPC(30): XFLD=51: YFLD=1:
6560 NUMDIG=10: INFLG=3: GOSUB 7120: IF ESCK=1 THEN 6640 ELSE TDMINX=S
6570 LOCATE 1,62,0: PRINT "X2=" SPC(14): XFLD=66: YFLD=1: GOSUB 7120: IF E
SCK=1 THEN 6640 ELSE TDMAXX=S
6580 IF TDMAXX<=TDMINX THEN GOSUB 30760: GOTO 6570
6590 LOCATE 1,47,0: PRINT "Y1=" SPC(30): XFLD=51: YFLD=1
6600 GOSUB 7120: IF ESCK=1 THEN 6640 ELSE TDMINY=S
6610 LOCATE 1,62,0: PRINT "Y2=" SPC(14): XFLD=66: YFLD=1: GOSUB 7120: IF E
SCK=1 THEN 6640 ELSE TDMAXY=S
6620 IF TDMAXY<=TDMINY THEN GOSUB 30760: GOTO 6610
6630 TDMINX=TDMINX: TMAXX=TMAXX: TDMINY=TDMINY: TMAXY=TMAXY: RETURN
6640 LOCATE 1,47,0: PRINT SPC(30): RETURN
6650 '
6660 ' SCREEN DUMP TO DISK
6670 LOCATE 1,47,0: PRINT "FILENAME >" SPC(24): NUMDIG=10: YFLD=1: XFLD=5
8
6680 INFLG=4: GOSUB 7120: IF ESCK=1 THEN 6800 ELSE FILENUM=1: FILETPE=
0
6690 FILENAM$=S$+".GPH": GOSUB 30090
6700 IF FERR$="OK" THEN CLOSE #FILENUM: LOCATE 1,47,0: PRINT SPC(30):
GOTO 6750
6710 LOCATE 1,47,0: PRINT FERR$ SPC(1) RT$: GOSUB 30760: INFLG=1: GOSUB
7120
6720 IF C$="Y" AND ESCK=0 THEN 6670 ELSE LOCATE 1,47,0: PRINT SPC(30
)
6730 IF ARC%=1 THEN COM$=FERR$+" on screen dump to disk": GOSUB 3041
0
6740 GOSUB 31530: RETURN
6750 ON ERROR GOTO 6790: DEF SEG=&HB800: BSAVE FILENAM$,0,&H4000: DEF
SEG
6770 IF ARC%=1 THEN COM$="Saved Graphics Screen in file "+FILENAM$:
GOSUB 30410
6780 ON ERROR GOTO 0: RETURN
6790 ON ERROR GOTO 0: GOSUB 31260: RESUME 6710

```

```

6800 LOCATE 1,47,0:PRINT SPC(30):RETURN
6810 '
6820 'TITLE ON GRAPH
6830 LOCATE 25,38:PRINT"REAL";:S$="IMAGINARY":FOR I=1 TO 9:T$=LEFT$(S$,1)
6840 S$=MID$(S$,2):LOCATE I+7,1,0:PRINT T$:NEXT I:RETURN
6845 '
6850 'PLOT GRAPH LABELS ROUTINE
6860 'Allows precise placement of a string in graphics mode
6870 '(X1,Y),(X2,Y) are graphics coordinates
6880 'S$ = string to be placed
6890 'string centered horizontally in the field
6900 LS=LEN(S$):C=INT((X2-X1)/8):IF LS>C THEN S$=LEFT$(S$,C)
6910 X=X1+(X2-X1+1)/2-8*LEN(S$)/2+4:Y=Y1+(Y2-Y1+1)/2-4
6920 WHILE S$<>""
6930 IF X>639 THEN 6950 ELSE C$=LEFT$(S$,1):S$=MID$(S$,2)
6940 GOSUB 6960:IF X>=0 THEN X=X+8
6950 WEND:RETURN
6960 'PLACE CHARACTER AT PARTICULAR COORDINATES
6970 A%(1)=8:A%(2)=8:DEF SEG=&HF000:OFFSET=8*ASC(C$)+&HFA6E
6980 FOR J=1 TO 8:B$(J)=HEX$(PEEK(OFFSET+J-1))
6990 IF LEN(B$(J))=1 THEN B$(J)="0"+B$(J)
7000 NEXT J
7010 FOR J=1 TO 4:A%(J+2)=VAL("&H"+B$(2*J)+B$(2*J-1)):NEXT J
7020 DEF SEG:PUT(X,Y),A%:RETURN
7030 '
7040 'INPUT ROUTINE for numbers, names, function keys, comments, &
Y/N answers
7050 'INFLG=1: gets Y/N answer, returns c$="Y" or c$="N"
7060 'INFLG=2: gets function key, returns c=asc(2nd char of key)
7070 'INFLG=3: gets number as string, checks if valid, returns s=number
7080 'INFLG=4: gets filename, returns s$=filename
7090 'INFLG=5: gets comment from user for archive, returns s$=comment
7100 'Calling routine specifies NUMDIG, XFLD, YFLD, and INFLG
7110 COLOR 6,0
7120 SXFLD=XFLD:S$="":C$="":T$="":N$="":ESCK=0:GOSUB 30030:IF SND%>1 THEN BEEP
7130 IF INFLG>2 THEN 7150
7140 C$=INKEY$:IF C$="" THEN 7140 ELSE 7170
7150 LOCATE YFLD,XFLD,1
7160 C$=INKEY$:IF C$="" THEN 7150
7170 IF C$=CHR$(27) THEN ESCK=1:RETURN 'escape key
7180 IF INFLG=2 THEN 7370
7190 IF LEN(C$)>1 THEN 7450 'invalid
7200 C=ASC(C$)
7210 IF INFLG=1 THEN 7320
7220 IF C=8 THEN 7500
7230 IF INFLG=5 THEN 7800 'comment
7240 IF INFLG=3 THEN 7390 'number

```

```

7250 'Name Routine
7260 IF C=13 THEN RETURN
7270 IF C=46 AND NUMDIG>10 THEN 7470 'okay
7280 IF C=91 OR C=92 OR C=93 OR C=124 THEN 7450 'invalid
7290 IF C=34 OR C=39 OR C=42 OR C=43 OR C=46 OR C=47 THEN 7450 'in
valid
7300 IF (C>32 AND C<59) OR (C>63 AND C<127) THEN 7470 ELSE 7450
7310 'Yes/No routine
7320 IF C=89 OR C=121 THEN C$='Y':RETURN
7330 IF C=78 OR C=110 THEN C$='N':RETURN
7340 IF C=13 THEN C$='N':RETURN
7350 GOTO 7450
7360 'Function Key
7370 IF LEN(C$)<2 THEN 7450 ELSE C$=RIGHT$(C$,1):C=ASC(C$)
7380 IF (C>58 AND C<69) OR (C>83 AND C<114) THEN RETURN ELSE 7450
7390 'Number Routine
7400 IF C=13 THEN 7540 'carriage return
7410 IF C>47 AND C<58 THEN 7470 'okay
7420 IF C=46 OR C=43 OR C=45 OR C=32 THEN 7470 'okay
7430 IF C=69 OR C=101 THEN C$=CHR$(69):GOTO 7470 'okay
7440 'Illegal character
7450 GOSUB 30760:BEEP:GOTO 7130
7460 'Character okay
7470 IF LEN(S$)>=NUMDIG THEN 7450
7480 LOCATE YFLD,XFLD:PRINT C$;:S$=S$+C$:XFLD=XFLD+1:GOTO 7130
7490 'Backspace
7500 XFLD=XFLD-1:LOCATE YFLD,XFLD:PRINT" ";:IF XFLD<=SXFLD THEN XFLD=SXFLD
7510 LNT=LEN(S$)-1:IF LNT<0 THEN LNT=0
7520 S$=MID$(S$,1,LNT):GOTO 7130
7530 'Number Check routine
7540 NUMCK=0:DECPT=0:EXPNT=0:SIGN=0
7550 IF S$="" THEN 7770 ELSE T$=LEFT$(S$,1)
7560 IF T$='+' OR T$='-' THEN N$=T$:S$=MID$(S$,2)
7570 IF T$='.' THEN N$=T$:DECPT=1:S$=MID$(S$,2)
7580 IF S$="" THEN 7790
7590 IF (T$='+' OR T$='-') AND INSTR(".",LEFT$(S$,1))=1 THEN DECPT=
1:T$=LEFT$(S$,1):N$=N$+T$:S$=MID$(S$,2)
7600 IF INSTR(DIGIT$,LEFT$(S$,1))=0 THEN 7790
7610 WHILE S$<>"":T$=LEFT$(S$,1):S$=MID$(S$,2)
7620 IF T$="" THEN 7680
7630 IF (T$='+' OR T$='-') AND RIGHT$(N$,1)<>"E" THEN NUMCK=1:GOTO
7680
7640 IF T$="." THEN IF DECPT=1 OR EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T
$:DECPT=1
7650 IF T$='+' OR T$='-' THEN IF SIGN=1 OR EXPNT=0 THEN NUMCK=1 EL
E N$=N$+T$:SIGN=1
7660 IF T$='E' THEN IF EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:EXPNT=1:D
ECPT=1
7670 IF T$<>"" AND T$<>"+ " AND T$<>"- " AND T$<>"E" THEN N$=N$+T$
7680 WEND:IF NUMCK=1 THEN 7790

```

```

7690 'Check for overflow
7700 D=0:N%=INSTR(N$,"E"): IF N%=0 THEN 7770
7710 IF N%=1 THEN 7790
7720 S$=LEFT$(N$,N%-1): C$=MID$(N$,N%+1): S=VAL(S$)
7730 IF S=0 THEN D=0 ELSE D=INT(LOG(ABS(S))/LOG(10))
7740 IF C$<>" " THEN D=D+VAL(C$)
7750 IF D<-37 OR D>37 THEN 7790
7760 'convert
7770 S=VAL(N$): RETURN
7780 'invalid
7790 NUMCK=1: GOSUB 30760: BEEP: LOCATE YFLD, SXFLD, 0: PRINT SPC(NUMDIG)
: XFLD= SXFLD: GOTO 7120
7800 'Comment routine
7810 IF C$=CHR$(13) THEN RETURN
7820 IF LEN(S$)>=NUMDIG THEN RETURN ELSE 7480
7830 '
30000 '***** SUBROUTINES *****
30010 '
30020 'CLEAR KBD BUFFER
30030 DEF SEG=0: POKE 1050, PEEK(1052): DEF SEG
30040 RETURN
30050 '
30060 'OPEN DISK FILE FOR INPUT/OUTPUT
30070 'Calling routine specifies filename$, filenum, fileapp, & file
tpe
30080 'fileapp=0 no append, fileapp=1 append, filetype=0 output =1 i
nput
30090 ON ERROR GOTO 30150: IF FILEAPP=0 THEN 30110 ELSE FILEAPP=0
30100 OPEN FILENAM$ FOR APPEND AS #FILENUM: GOTO 30140
30110 IF FILETPE=1 THEN 30130
30120 OPEN FILENAM$ FOR OUTPUT AS #FILENUM: GOTO 30140
30130 OPEN FILENAM$ FOR INPUT AS #FILENUM
30140 FERR$="OK"
30145 ON ERROR GOTO 0: RETURN
30150 GOSUB 31260: RESUME 30145
30170 '
30180 'FILE DIRECTORY SUBROUTINE
30190 ON ERROR GOTO 30280
30200 COLOR 12,0: CLS: GOSUB 30030: LOCATE 1,18,0: IF SND%>1 THEN BEEP
30210 PRINT "WHICH DRIVE? (A, B, OR AB for both) >";: COLOR 6,0: INPU
T"", COM$
30220 PRINT: COLOR 3,0
30230 IF COM$="a:" OR COM$="A:" OR COM$="a" OR COM$="A" THEN PRINT "
A: \": FILES" A: *. *": GOTO 30310
30240 IF COM$="b:" OR COM$="B:" OR COM$="b" OR COM$="B" THEN PRINT "
B: \": FILES" B: *. *": GOTO 30310
30250 IF COM$="ab:" OR COM$="AB:" OR COM$="ab" OR COM$="AB" THEN 30
260 ELSE 30270
30260 PRINT "A: \": FILES" A: *. *": PRINT "B: \": FILES" B: *. *": GOTO 30310
30270 GOSUB 30760: GOTO 30200
30280 GOSUB 31260: GOSUB 30760: COLOR 15,0: PRINT: PRINT SPC(30) FERR$:

```

```

COLOR 7,0
30290 PRINT:PRINT SPC(31) RT$:INFLG=1:GOSUB 7120
30300 IF C$="Y" THEN RESUME 30200 ELSE RESUME 30310
30310 ON ERROR GOTO 0
30320 YFLD=24:GOSUB 31680:LOCATE 24,28,0:COLOR 12,0:PRINT"KILL ANY
FILES? (Y/N)";:INFLG=1:GOSUB 7120:IF C$="N" OR ESCK=1 THEN RETURN
30330 GOSUB 31680:LOCATE 24,32,0:PRINT"FILENAME >";:COLOR 6,0
30340 YFLD=24:XFLD=43:NUMDIG=14:INFLG=4:GOSUB 7110:IF ESCK=1 THEN R
ETURN
30350 FILENAM$=S$:GOSUB 31530:IF FERR$="OK" THEN 30320 ELSE GOSUB 3
1680:GOSUB 30760
30360 COLOR 15,0:LOCATE 24,13,0:PRINT FERR$;:LOCATE 24,47,0:COLOR 7
,0:PRINT RT$;
30370 INFLG=1:GOSUB 7120:IF C$="N" OR ESCK=1 THEN RETURN ELSE COLOR
12,0:GOTO 30330
30380 '
30390 'PROGRAM WRITE TO ARCHIVE
30400 'Calling routine provides COM$
30410 FILENAM$=ARCNAM$:FILENUM=2:FILEAPP=1:FILETPE=0
30420 GOSUB 30090:IF FERR$<>"OK" THEN CLOSE #FILENUM:GOTO 30450 'op
en file
30430 ON ERROR GOTO 30490:PRINT #FILENUM,COM$
30440 CLOSE #FILENUM:ON ERROR GOTO 0:RETURN
30450 CLS:COLOR 15,0:LOCATE 10,28,0:PRINT"ERROR ARCHIVING SESSION"
30460 LOCATE 12,28,0:PRINT FERR$:COLOR 7,0:GOSUB 30760
30470 LOCATE 14,30,0:PRINT RT$:INFLG=1:GOSUB 7120
30480 IF C$="Y" THEN 30420 ELSE RETURN
30490 ON ERROR GOTO 0:GOSUB 31260:RESUME 30450
30500 '
30510 'FILENAME INFORMATION & DIRECTORY ROUTINE
30520 COLOR 3,0:CLS
30530 LOCATE 2,15,0:PRINT"A filename is 1-8 letters/digits. A file
name may"
30540 LOCATE 3,15,0:PRINT"be preceded by a disk drive letter and a
colon,"
30550 LOCATE 4,15,0:PRINT"otherwise the current logged drive is use
d."
30560 IF FILETPE=0 THEN COLOR 15,0:LOCATE 6,8,0:PRINT"WARNING! Usi
ng an existing filename will erase the existing file."
30570 COLOR 3,0:LOCATE 8,24,0:PRINT"Would you like to see your"
30580 COLOR 12,0:LOCATE 9,26,0:PRINT"File Directory? (Y/N)"
30590 INFLG=1:GOSUB 7120:IF C$="Y" THEN GOSUB 30190
30600 RETURN
30610 '
30620 'INPUT FILENAME ROUTINE
30630 'calling routine supplies filext$, filenum, filetype, fileapp
30640 'leaves file open if OK, closes if FO=1
30650 CLS:COLOR 3,0:LOCATE 2,24,0:PRINT"Enter filename including dr
ive."
30660 LOCATE 3,20,0:PRINT"File will be given the extension "FILEXT$
30670 LOCATE 5,30,0:COLOR 12,0:PRINT"FILENAME >"

```

```

30680 XFLD=41:YFLD=5:NUMDIG=10:INFLG=4:GOSUB 7110:IF ESCK=1 THEN RE
TURN
30690 FILENAM$=S$+FILEXT$:GOSUB 30090
30700 IF FERR$="OK" THEN RETURN ELSE CLOSE #FILENUM
30710 COLOR 15,0:GOSUB 30760:LOCATE 7,28,0:PRINT FERR$:COLOR 7,0
30720 LOCATE 9,30,0:PRINT RT$:INFLG=1:GOSUB 7120
30730 IF C$="Y" THEN 30650 ELSE RETURN
30740 '
30750 'FLUB SOUND SUBROUTINE
30760 IF SND%<1 THEN RETURN ELSE SOUND 1000,2:SOUND 1180,3:BEEP:RET
URN
30770 '
30780 'PRESS ANY KEY ROUTINE
30790 'Caller specifies XFLD & YFLD coordinates of message
30800 COLOR 7,0:LOCATE YFLD,XFLD,0:PRINT"Press any key.....";:GOS
UB 30030
30810 IF SND%>1 THEN BEEP
30820 C$=INKEY$:IF C$="" THEN 30820
30830 IF C$=CHR$(27) THEN ESCK=1 ELSE ESCK=0
30840 RETURN
30850 COLOR 7,0:PRINT CHR$(13) CHR$(13) "SPC(29)"Press any key.....
."";:GOSUB 30030:GOTO 30810
30860 '
30870 'INPUT/PRINT MATRIX ROUTINE
30880 'Caller specifies MAT, XFLD, YFLD, NUMDIG, P=1 PRINT
30890 COLOR 12,0:LOCATE YFLD,XFLD,0:PRINT COM$:YFLD=YFLD+2:XFLD=1:I
NFLG=3
30900 FOR I=1 TO ROW%:XOFF=6:NC=0:COLOR 12,0:LOCATE YFLD,XFLD:PRINT
"R";
30910 LOCATE YFLD,XFLD+1:PRINT I;:COLOR 6,0:XFLD=6:FOR J=1 TO COL%:
NC=NC+1
30920 IF P=1 THEN 30980 ELSE GOSUB 7110:IF ESCK=1 THEN RETURN
30930 IF MAT=1 THEN A(J)=S:GOTO 31040
30940 IF MAT=2 THEN B(J)=S:GOTO 31040
30950 IF MAT=3 THEN C(I)=S:GOTO 31040
30960 IF MAT=4 THEN D(I)=S:GOTO 31040
30980 LOCATE YFLD,XFLD,0
30990 IF MAT=1 THEN PRINT A(J);:GOTO 31040
31000 IF MAT=2 THEN PRINT B(J);:GOTO 31040
31010 IF MAT=3 THEN PRINT C(I);:GOTO 31040
31020 IF MAT=4 THEN PRINT D(I);:GOTO 31040
31035 IF MAT=6 THEN PRINT SPO(I,J);
31040 XFLD=NC*(NUMDIG+2)+XOFF:IF XFLD>70 THEN YFLD=YFLD+1:XFLD=10:X
OFF=10:NC=0:IF YFLD>22 THEN GOSUB 31070
31050 NEXT J:YFLD=YFLD+1:IF YFLD>22 THEN GOSUB 31070
31060 XFLD=1:NEXT I:RETURN
31070 SXFLD=XFLD:XFLD=28:YFLD=24:GOSUB 30800:YFLD=3:XFLD=SXFLD:COLO
R 6,0:CLS:RETURN
31080 '
31090 'READ MATRIX FROM DISK FILE
31100 'Caller specifies MAT, FILENUM, ROW%, COL%, & opens file

```

```

31110 ON ERROR GOTO 31220
31120 FOR I=1 TO ROW%:FOR J=1 TO COL%:IF EOF(FILENUM) THEN 31210
31130 IF MAT=0 THEN INPUT #FILENUM,KDIM(I):GOTO 31200
31140 IF MAT=1 THEN INPUT #FILENUM,A(J):GOTO 31200
31150 IF MAT=2 THEN INPUT #FILENUM,B(J):GOTO 31200
31160 IF MAT=3 THEN INPUT #FILENUM,C(I):GOTO 31200
31170 IF MAT=4 THEN INPUT #FILENUM,D(I):GOTO 31200
31190 IF MAT=6 THEN INPUT #FILENUM,OP(I,J)
31200 NEXT J,I:FERR$="OK":RETURN
31210 FERR$="END OF FILE BEFORE FINISHED":RETURN
31220 ON ERROR GOTO 0:GOSUB 31260:RESUME 31230
31230 RETURN
31240 '
31250 'ERROR CHECKING ROUTINE
31260 IF ERR=64 THEN FERR$="BAD FILE NAME":RETURN
31270 IF ERR=70 THEN FERR$="DISK WRITE PROTECT":RETURN
31280 IF ERR=53 THEN FERR$="FILE NOT FOUND":RETURN
31290 IF ERR=67 THEN FERR$="TOO MANY FILES ON DISK":RETURN
31300 IF ERR=55 THEN FERR$="FILE ALREADY IN USE":RETURN
31310 IF ERR=71 THEN FERR$="DISK NOT READY":RETURN
31320 IF ERR=61 THEN FERR$="DISK FULL":RETURN
31330 IF ERR=72 THEN FERR$="BAD DISK MEDIA":RETURN
31340 FERR$="DISK I/O ERROR":RETURN
31350 '
31360 'SAVE MATRIX ON DISK
31370 'Caller specifies MAT, FILENUM, ROW%, COL% & opens file
31380 ON ERROR GOTO 31480
31390 FOR I=1 TO ROW%:FOR J=1 TO COL%
31400 IF MAT=0 THEN PRINT #FILENUM,KDIM(I);:GOTO 31470
31410 IF MAT=1 THEN PRINT #FILENUM,A(J);:GOTO 31470
31420 IF MAT=2 THEN PRINT #FILENUM,B(J);:GOTO 31470
31430 IF MAT=3 THEN PRINT #FILENUM,C(I);:GOTO 31470
31440 IF MAT=4 THEN PRINT #FILENUM,D(I);:GOTO 31470
31460 IF MAT=6 THEN PRINT #FILENUM,OP(I,J);
31470 NEXT J,I:FERR$="OK":ON ERROR GOTO 0:RETURN
31480 ON ERROR GOTO 0:GOSUB 31260:RESUME 31490
31490 RETURN
31500 '
31510 'KILL FILE ROUTINE
31520 'Caller specifies FILENAM$
31530 ON ERROR GOTO 31550:KILL FILENAM$:FERR$="OK"
31540 ON ERROR GOTO 0:RETURN
31550 GOSUB 31260:RESUME 31540
31570 '
31580 'CHANGE MATRIX ELEMENT
31590 'Caller specifies MAT, ROW1%, COL1%, & S (the new value)
31600 IF MAT=0 THEN KDIM(ROW1%)=S:RETURN
31610 IF MAT=1 THEN A(COL1%)=S:RETURN
31620 IF MAT=2 THEN B(COL1%)=S:RETURN
31630 IF MAT=3 THEN C(ROW1%)=S:RETURN
31640 IF MAT=4 THEN D(ROW1%)=S:RETURN

```

```

31660 '
31670 'ERASE ANY LINE ROUTINE
31680 LOCATE YFLD,1,0: PRINT SPC(79);: RETURN
31690 '
31700 'WRITE BLANK TO ARCHIVE
31710 COM$="": GOSUB 30410: RETURN
31720 '
32300 'NO MODEL/MATRIX ERROR ROUTINE
32310 GOSUB 30760: CLS: COLOR 15,0: LOCATE 10,28,0: PRINT COM$: XFLD=28:
YFLD=12: GOSUB 30800: RETURN
32320 '
32330 'ERASE EXISTING ERROR ROUTINE
32340 GOSUB 30760: CLS: COLOR 15,0: LOCATE 10,24,0: PRINT COM$: COLOR 7,
0: INFLG=1: GOSUB 7120: RETURN
32350 '
32430 'MODIFY ANY MATRIX ROUTINE
32440 CLS: XFLD=1: YFLD=3: NUMDIG=10: P=1: GOSUB 30890: ROW1%=1
32450 COLOR 12,0: LOCATE 25,20,0: PRINT"ELEMENT COLUMN >" SPC(12);
32470 XFLD=37: YFLD=25: NUMDIG=10: INFLG=3: GOSUB 7110: IF ESCK=1 THEN R
ETURN
32540 IF S>COL% OR S<1 THEN GOSUB 30760: GOTO 32450
32550 COL1%=S: LOCATE 25,37,0: PRINT COL1% SPC(6);: LOCATE 25,41,0: COL
OR 12,0: PRINT"VALUE >";: XFLD=49: YFLD=25
32560 GOSUB 7110: IF ESCK=1 THEN RETURN ELSE GOSUB 31600: GOTO 32440
32590 '
32700 GOSUB 30760: CLS: COLOR 15,0: LOCATE 10,28,0: PRINT COM$: COLOR 7,
0
32710 LOCATE 12,33,0: PRINT RT$: INFLG=1: GOSUB 7120: RETURN
32720 '
32770 'ROOT FINDING ROUTINE
32780 'N=order of polynomial, C=matrix of coefficients
32790 IREV=1: NC=N+1: FOR I=1 TO NC: D(I)=C(I): NEXT I: P=0: Q=0: R=0
32795 IF N<5 THEN 34000
32800 IF D(1)<>0 THEN 32820 ELSE NC=NC-1
32810 SPO(NC,1)=0: SPO(NC,2)=0: FOR I=1 TO NC: D(I)=D(I+1): NEXT I: GOTO
32800
32820 IF (NC-1)=0 THEN RETURN
32830 IF (NC-2)=0 THEN R=-D(1)/D(2): GOTO 33050
32840 IF (NC-3)=0 THEN P=D(2)/D(3): Q=D(1)/D(3): GOTO 33070
32850 IF (ABS(D(NC-1)/D(NC))-ABS(D(2)/D(1)))>=0 THEN 32900
32860 IREV=-IREV: FOR I=1 TO NC/2: NL=NC+1-I: TEMP=D(NL): D(NL)=D(I): D(
I)=TEMP: NEXT I
32870 IF Q=0 THEN P=0: GOTO 32890
32880 P=P/Q: Q=1/Q
32890 IF R<>0 THEN R=1/R
32900 E=5E-10: SG(NC)=D(NC): C(NC)=D(NC): SG(NC+1)=0: C(NC+1)=0: NP=NC-1
32910 FOR J=1 TO 1000: FOR II=1 TO NP: I=NC-II: SG(I)=D(I)+R*SG(I+1)
32920 C(I)=SG(I)+R*C(I+1): NEXT II
32930 IF (ABS(SG(1)/D(1))-E)<=0 THEN 33050
32940 IF C(2)=0 THEN R=R+1 ELSE R=R-SG(1)/C(2)
32950 FOR II=1 TO NP: I=NC-II

```

```

32960 SG(I)=D(I)-P*SG(I+1)-Q*SG(I+2): C(I)=SG(I)-P*C(I+1)-Q*C(I+2): N
EXT I I
32970 IF D(2)<>0 THEN 32990
32980 IF ABS(SG(2)/D(1))-E<=0 THEN 33000 ELSE 33010
32990 IF ABS(SG(2)/D(2))-E>0 THEN 33010
33000 IF ABS(SG(1)/D(1))-E<=0 THEN 33070
33010 CBAR=C(2)-SG(2): D=C(3)*2-CBAR*C(4)
33020 IF D=0 THEN P=P-2: Q=Q*(Q+1): GOTO 33040
33030 P=P+(SG(2)*C(3)-SG(1)*C(4))/D: Q=Q+(-SG(2)*CBAR+SG(1)*C(3))/D
33040 NEXT J: E=E*10: GOTO 32910
33050 NC=NC-1: SPO(NC,2)=0: IF IREV<0 THEN SPO(NC,1)=1/R ELSE SPO(NC,
1)=R
33060 FOR I=1 TO NC: D(I)=SG(I+1): NEXT I: GOTO 32820
33070 NC=NC-2: IF IREV<0 THEN QP=1/Q: PP=P/(Q*2): GOTO 33090
33080 QP=Q: PP=P/2
33090 TEMP=(PP)*2-QP: IF TEMP>=0 THEN 33110
33100 SPO(NC+1,1)=-PP: SPO(NC,1)=-PP: SPO(NC+1,2)=SQR(-TEMP): SPO(NC,2
)=-SPO(NC+1,2): GOTO 33130
33110 IF PP=0 THEN SPO(NC+1,1)=-SQR(TEMP) ELSE SPO(NC+1,1)=(-PP/ABS
(PP))*(ABS(PP)+SQR(TEMP))
33120 SPO(NC+1,2)=0: SPO(NC,1)=QP/SPO(NC+1,1): SPO(NC,2)=0
33130 FOR I=1 TO NC: D(I)=SG(I+2): NEXT I: GOTO 32820
34000 IF N=1 THEN SPO(N,1)=-D(N): SPO(N,2)=0: RETURN
34010 IF N=3 THEN 34070
34020 IF N=4 THEN 34200
34030 SR=D(2)*2-4*D(1): SI=0: N=2: GOSUB 34610
34040 IF RI<>0 THEN 34060 ELSE SPO(1,2)=0: SPO(2,2)=0
34050 SPO(1,1)=(-D(2)+RR)/2: SPO(2,1)=(-D(2)-RR)/2: RETURN
34060 SPO(1,1)=-D(2)/2: SPO(2,1)=SPO(1,1): SPO(1,2)=RI/2: SPO(2,2)=-RI
/2: RETURN
34070 A=(3*D(2)-D(3)*2)/3: B=(2*D(3)*3-9*D(3)*D(2)+27*D(1))/27
34080 SR=B*2/4+A*3/27: SI=0: N=2: GOSUB 34610
34090 QTR=RR: QTI=RI: SR=QTR-B/2: SI=QTI: N=3
34100 IF SR<0 AND SI=0 THEN SR=-SR: GOSUB 34610: RR=-RR: GOTO 34120
34110 GOSUB 34610
34120 QAR=RR: QAI=RI: SR=-QTR-B/2: SI=-QTI
34130 IF SR<0 AND SI=0 THEN SR=-SR: GOSUB 34610: RR=-RR: GOTO 34150
34140 GOSUB 34610
34150 QBR=RR: QBI=RI: SR=-3: SI=0: N=2: GOSUB 34610
34160 SPO(1,1)=QAR+QBR: SPO(1,2)=QAI+QBI: QTR=(QAR-QBR)/2: QTI=(QAI-QB
I)/2
34170 SR=FNCRM(QTR,QTI,RR,RI): SI=FNCIM(QTR,QTI,RR,RI)
34180 SSR=-(QAR+QBR)/2: SSI=-(QAI+QBI)/2: SPO(2,1)=SSR+SR: SPO(2,2)=SS
I+SI: SPO(3,1)=SSR-SR: SPO(3,2)=SSI-SI
34190 FOR I=1 TO 3: SPO(I,1)=SPO(I,1)-D(3)/3: NEXT I: RETURN
34200 A=D(4): B=D(3): C=D(2): D=D(1)
34210 FOR I=1 TO 4: SPO(I,1)=0: SPO(I,2)=0: NEXT I: PI=3.1415926#: PIO2=
1.5707963#
34220 QP=-B: QQ=A*C-4*D: QR=-(A*2)*D+4*B*D-C*2
34230 QCA=(3*QQ-QP*2)/3: QCB=(2*QP*3-9*QP*QQ+27*QR)/27
34240 QTEMP=QCB*2/4+QCA*3/27: N=2: SR=QTEMP: SI=0

```

```

34250 GOSUB 34610: QTR=RR: QTI=RI
34260 SR=QTR-QCB/2: SI=QTI: N=3
34270 IF SR<0 AND SI=0 THEN SR=-SR: GOSUB 34610: RR=-RR: GOTO 34290
34280 GOSUB 34610
34290 QCAAR=RR: QCAAI=RI: SR=-QTR-QCB/2: SI=-QTI
34300 IF SR<0 AND SI=0 THEN SR=-SR: GOSUB 34610: RR=-RR: GOTO 34320
34310 GOSUB 34610
34320 YR=QCAAR+RR-QP/3: YI=QCAAI+RI
34330 SR=Ac2/4-B+YR: SI=YI: N=2: GOSUB 34610
34340 QRR=RR: QRI=RI
34350 IF QRR=0 AND QRI=0 THEN 34450
34360 QTEMP=4*A*B-8*C-Ac3
34370 QTR=FNCRD(QTEMP, 0, 4*QRR, 4*QRI)
34380 QTI=FNCID(QTEMP, 0, 4*QRR, 4*QRI)
34390 QRTR=FNCRM(QRR, QRI, QRR, QRI)
34400 QRTI=FNCIM(QRR, QRI, QRR, QRI)
34410 SR=3*Ac2/4-QRTR-2*B+QTR: N=2: SI=-QRTI+QTI: GOSUB 34610
34420 QDR=RR: QDI=RI
34430 SR=3*Ac2/4-QRTR-2*B-QTR: N=2: SI=-QRTI-QTI: GOSUB 34610
34440 QER=RR: QEI=RI: GOTO 34500
34450 QTR=FNCRM(YR, YI, YR, YI)-4*D
34460 QTI=FNCIM(YR, YI, YR, YI)
34470 SR=QTR: SI=QTI: N=2: GOSUB 34610: QTR=2*RR: QTI=2*RI
34480 SR=QTR+3*Ac2/4-2*B: SI=QTI: GOSUB 34610: QDR=RR: QDI=RI
34490 SR=-QTR+3*Ac2/4-2*B: SI=-QTI: GOSUB 34610: QER=RR: QEI=RI
34500 SPO(1,1)=-A/4+QRR/2+QDR/2
34510 SPO(1,2)=QRI/2+QDI/2
34520 SPO(2,1)=-A/4+QRR/2-QDR/2
34530 SPO(2,2)=QRI/2-QDI/2
34540 SPO(3,1)=-A/4-QRR/2+QER/2
34550 SPO(3,2)=-QRI/2+QEI/2
34560 SPO(4,1)=-A/4-QRR/2-QER/2
34570 SPO(4,2)=-QRI/2-QEI/2
34580 RETURN
34590 '
34600 'This subroutine finds the Nth root of a complex number.
34610 PI=3.1415926#: PIO2=1.5707963#
34615 IF SI=0 AND SR=0 THEN RI=0: RR=0: RETURN
34620 IF SI<0 AND SR<0 THEN ANGLE=-PI+ATN(SI/SR): GOTO 34700
34630 IF SI<0 AND SR=0 THEN ANGLE=-PIO2: GOTO 34700
34640 IF SI<0 AND SR>0 THEN ANGLE=-ATN(-SI/SR): GOTO 34700
34650 IF SI=0 AND SR<0 THEN ANGLE=PI: GOTO 34700
34660 IF SI=0 AND SR>0 THEN ANGLE=0: GOTO 34700
34670 IF SI>0 AND SR<0 THEN ANGLE=PI-ATN(-SI/SR): GOTO 34700
34680 IF SI>0 AND SR=0 THEN ANGLE=PIO2: GOTO 34700
34690 IF SI>0 AND SR>0 THEN ANGLE=ATN(SI/SR)
34700 NR=SQR(SIc2+SRc2): RR=NRc(1/N)*COS(ANGLE/N): RI=NRc(1/N)*SIN(ANGLE/N)
34710 RETURN

```

C.3 FREQ 3/6/85

```

10 COMMON HELP%,HELP$,ARC%,SND%,QW,ARCNAM$: OPTION BASE 1:DEFINT I,J
20 RL$="-----FREQUENCY RESPONSE-----"
40 DIM A(21),B(21),OP(1004,3),PX(1004),PY(1004),SDJ(21),SDI(21),DB(
101,2),SNG(10)
50 DIM C(2,21),D(2,21),SG(21),SW(21,2),B$(10),A%(10),KDIM(7),SP0(21
,2)
55 DEF FNPSC(PS,PXS,PNS,PXO,PNO)=INT(PXO-(PXO-PNO)/(PXS-PNS)*(PXS-P
S)+.5)
56 DEF FNLOG10(PONT)=LOG(PONT)/LOG(10)
58 IF QW<1 THEN HELP%=0:HELP$="OFF":ARC%=0:SND%=1
60 MAXDIM=20:MAXPTS=1000:DIGIT$="1234567890":RT$="RETRY? (Y/N)"
62 DEF FNCRD(XR,XI,YR,YI)=(XR*YR+XI*YI)/(YR^2+YI^2)
64 DEF FNCID(XR,XI,YR,YI)=(XI*YR-XR*YI)/(YR^2+YI^2)
66 DEF FNCRM(XR,XI,YR,YI)=(XR*YR-XI*YI)
68 DEF FNCIM(XR,XI,YR,YI)=(XR*YI+XI*YR)
70 EM=0: IF ARC%=1 THEN GOSUB 31710:COM$=RL$:GOSUB 30410:GOSUB 31710
80 SCREEN 0,1:CLS:WIDTH 80:COLOR 14,0:LOCATE 3,27,0:PRINT"FREQUENCY
RESPONSE"
90 COLOR 10,0:LOCATE 6,23,0:PRINT"<F1>.....ENTER NEW MODEL"
100 LOCATE 7,23,0:PRINT"<F2>.....RECALL OLD MODEL"
110 LOCATE 8,23,0:PRINT"<F3>.....MODIFY CURRENT MODEL"
120 LOCATE 9,23,0:PRINT"<F4>.....STORE MODEL ON DISK"
130 LOCATE 10,23,0:PRINT"<F5>.....ENTER INPUT PARAMETERS"
140 LOCATE 11,23,0:PRINT"<F6>.....CALCULATE DATA"
150 LOCATE 12,23,0:PRINT"<F7>.....GRAPH DATA"
160 LOCATE 13,23,0:PRINT"<F8>.....LIST/SAVE DATA"
170 LOCATE 14,23,0:PRINT"<F9>.....FILE DIRECTORY"
180 LOCATE 15,23,0:PRINT"<ESC>.....EXIT TO MAIN MENU"
190 LOCATE 16,23,0:PRINT"<Alt-F1>....HELP LEVEL"
200 LOCATE 17,23,0:PRINT"<Alt-F2>....SOUND LEVEL"
210 COLOR 14,0:LOCATE 16,46,0:PRINT HELP$:LOCATE 17,46,0:PRINT SND%
220 COLOR 10,0:LOCATE 18,23,0:PRINT"<Alt-F3>....ARCHIVE/COMMENTS"
230 COLOR 14,0:LOCATE 21,24,0:PRINT"PRESS DESIRED FUNCTION KEY"
240 INFLG=2:GOSUB 7120: IF ESCK=1 AND QW=1 THEN CHAIN"DACS"
245 IF ESCK=1 AND QW<>1 THEN 382      'system
250 IF C=59 THEN GOSUB 630:GOTO 80    'new model
260 IF C=60 THEN GOSUB 1100:GOTO 80   'recall model
270 IF C=61 THEN GOSUB 1700:GOTO 80   'modify
280 IF C=62 THEN GOSUB 940:GOTO 80    'store model
290 IF C=63 THEN GOSUB 1340:GOTO 80   'input params
300 IF C=64 THEN GOSUB 1820:GOTO 80   'run
310 IF C=65 THEN GOSUB 2820:GOTO 80   'graph
320 IF C=66 THEN GOSUB 3590:GOTO 80   'list
330 IF C=67 THEN GOSUB 30190:GOTO 80  'files
340 IF C=104 THEN GOSUB 590:GOTO 210  'help
350 IF C=105 THEN GOSUB 550:GOTO 210  'help
360 IF C=106 THEN GOSUB 390:GOTO 80   'archive
370 GOSUB 30760:GOTO 240
380 '

```

```

382 CLEAR:WIDTH 80:SCREEN 0,0:COLOR 7,0:CLOSE:END
383 '
385 'ARCHIVE
390 IF ARC%=1 THEN GOSUB 420:RETURN
395 GOSUB 420:IF ARC%=0 THEN RETURN ELSE GOSUB 31710:COM$=RL$:GOSUB
  30410:GOSUB 31710:RETURN
400 '
410 'ARCHIVE SESSION / ADD COMMENTS
420 COLOR 12,0:CLS:IF ARC%=1 THEN 470 ELSE LOCATE 2,27,0
430 PRINT"ARCHIVE SESSION? (Y/N)":INFLG=1:GOSUB 7120:IF ESCK=1 OR
  C$="N" THEN RETURN
440 FILETPE=0:IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
450 FILEXT$=".ARC":FILENUM=2:GOSUB 30650:CLOSE #FILENUM
460 IF ESCK=1 OR FERR$<>"OK" THEN RETURN ELSE ARCNAM$=FILENAM$:ARC%
  =1:CLS
470 COLOR 12,0:LOCATE 2,28,0:PRINT"ENTER COMMENTS? (Y/N)":INFLG=1:
  GOSUB 7120
480 IF C$="N" OR ESCK=1 THEN RETURN ELSE NUMDIG=78:INFLG=5:XFLD=1
490 CLS:LOCATE 2,19,0:YFLD=4
500 PRINT"ENTER COMMENTS BELOW....PRESS ESC TO QUIT":PRINT
510 GOSUB 7110:IF ESCK=1 THEN GOSUB 31710:RETURN
520 COM$=S$:GOSUB 30410:IF FERR$<>"OK" THEN RETURN ELSE YFLD=YFLD+1
  :XFLD=1:IF YFLD>24 THEN 490 ELSE 510
530 '
540 'SOUND LEVEL
550 SND%=SND%+1:IF SND%>2 THEN SND%=0
560 RETURN
570 '
580 'SET HELP LEVEL
590 IF HELP%=1 THEN HELP%=0:HELP$="OFF":RETURN
600 HELP%=1:HELP$="ON ":RETURN
610 '
620 'ENTER NEW MODEL
630 IF EM=1 THEN COM$="ERASE EXISTING MODEL? (Y/N)":GOSUB 32340:IF
  C$="N" OR ESCK=1 THEN RETURN
640 CLS:IF HELP%=0 THEN 755 ELSE COLOR 3,0
650 LOCATE 2,12,0:PRINT"This program calculates the frequency respo
  nse of a rational"
660 LOCATE 3,12,0:PRINT"transfer function G(s) of the form"
670 LOCATE 5,30,0:COLOR 15,0:PRINT"G(s) = KN(s) / D(s)":COLOR 3,0
690 LOCATE 7,12,0:PRINT"You will be prompted for the polynomials N(
  s) & D(s). "
700 LOCATE 8,12,0:PRINT"The coefficient of the highest order term i
  s assumed unity."
710 LOCATE 9,12,0:PRINT"Enter the coefficients in ascending order.
  Example:"
720 LOCATE 11,20,0:COLOR 15,0:PRINT"G(s) = (s + 3)/(s**3 + 2s**2 +
  9s + 3)"
730 LOCATE 13,30,0:PRINT"N(s) = 3":LOCATE 14,30,0:PRINT"D(s) = 3
  9 2":COLOR 3,0
740 LOCATE 16,12,0:PRINT"The coefficient of the highest order term

```

is not entered."

```

750 GOSUB 30850: IF ESCK=1 THEN 910 ELSE CLS
755 GOSUB 760: IF ESCK=1 THEN 910 ELSE GOSUB 850: IF ESCK=1 THEN 910
756 EM=1: IF ARC%=1 THEN GOSUB 31710: COM$="-ENTER NEW MODEL-": GOSUB
30410
757 RETURN
760 COLOR 12,0: LOCATE 10,30,0: PRINT"N(s) DIMENSION >"SPC(20)
770 XFLD=47: YFLD=10: INFLG=3: NUMDIG=12: GOSUB 7110: IF ESCK=1 THEN RET
URN
780 S=INT(S): IF S<=MAXDIM AND S>=0 THEN KDIM(1)=S: GOTO 800
790 GOSUB 30760: COLOR 15,0: LOCATE 12,18,0: PRINT"ERROR! Value must
lie in the range 0 to"MAXDIM: GOTO 760
800 COLOR 12,0: LOCATE 14,30,0: PRINT"D(s) DIMENSION >"SPC(20)
810 XFLD=47: YFLD=14: INFLG=3: NUMDIG=12: GOSUB 7110: IF ESCK=1 THEN RET
URN
820 S=INT(S): IF S<=MAXDIM AND S>=0 THEN KDIM(2)=S: RETURN
830 COLOR 15,0: LOCATE 16,18,0: PRINT"ERROR! Value must lie in the r
ange 0 to"MAXDIM: GOSUB 30760: GOTO 800
850 CLS: NUMDIG=10: INFLG=3: P=0: IF HELP%=0 THEN 870 ELSE COLOR 3,0
860 LOCATE 1,3,0: PRINT"ENTER POLYNOMIALS IN ASCENDING ORDER PRESSIN
G <ENTER> AFTER EACH ELEMENT.";
870 A(KDIM(1)+1)=1: B(KDIM(2)+1)=1: IF KDIM(1)=0 THEN 876
875 MAT=1: COM$="N(s) POLYNOMIAL": XFLD=1: YFLD=3: ROW%=1: COL%=KDIM(1):
GOSUB 30890: IF ESCK=1 THEN RETURN
876 IF KDIM(2)=0 THEN RETURN ELSE MAT=2
880 COM$="D(S) POLYNOMIAL": XFLD=1: YFLD=9: ROW%=1: COL%=KDIM(2): GOSUB
30890: RETURN
910 COM$="MODEL NOT ENTERED!!": GOSUB 32310: EM=0: RETURN
920 '
930 ' SAVE MODEL ON DISK
940 IF EM=0 THEN COM$="NO MODEL TO SAVE!!": GOSUB 32310: RETURN
950 FILETPE=0: IF HELP%=1 THEN GOSUB 30510 ELSE CLS: GOSUB 30580
960 FILEXT$=".TMD": FILENUM=1: GOSUB 30650
970 IF ESCK=0 AND FERR$="OK" THEN 990
980 COM$="MODEL NOT SAVED!!": GOSUB 32310: RETURN
990 SFILE$=FILENAM$: CLS: COLOR 3,0: LOCATE 10,30,0: PRINT"SAVING MODEL
....."
1000 IF ARC%=1 THEN GOSUB 31710: COM$="-STORE MODEL ON DISK-": GOSUB
30410: FILENUM=1
1010 MAT=0: ROW%=2: COL%=1: GOSUB 31380: IF FERR$<>"OK" THEN 1050
1020 MAT=1: ROW%=1: COL%=KDIM(1)+1: GOSUB 31380: IF FERR$<>"OK" THEN 10
50
1030 MAT=2: ROW%=1: COL%=KDIM(2)+1: GOSUB 31380: IF FERR$<>"OK" THEN 10
50
1035 CLOSE#FILENUM: IF ARC%=1 THEN COM$="Model saved under filename"
+SFILE$: GOSUB 30410
1040 RETURN
1050 COM$=FERR$: GOSUB 32700: FILENAM$=SFILE$: GOSUB 31530: IF C$="Y" A
ND ESCK=0 THEN 950
1060 IF ARC%=1 THEN COM$=FERR$: GOSUB 30410
1070 GOTO 980

```

```

1080 '
1090 'RECALL MODEL ROUTINE
1100 IF EM=1 THEN COM$='ERASE EXISTING MODEL? (Y/N)':GOSUB 32340:IF C$="N" OR ESCK=1 THEN RETURN
1110 FILETPE=1: IF HELP%=1 THEN GOSUB 30520 ELSE CLS:GOSUB 30580
1120 FILEXT$=".TMD":FILENUM=1:GOSUB 30650
1130 IF ESCK=0 AND FERR$="OK" THEN SFILE$=FILENAM$:GOTO 1150
1140 COM$="MODEL NOT RECALLED!!":GOSUB 32310:RETURN
1150 CLS:COLOR 3,0:LOCATE 10,30,0:PRINT"RECALLING MODEL....."
1160 IF ARC%=1 THEN GOSUB 31710:COM$="-RECALL MODEL-":GOSUB 30410:FILENUM=1
1170 ROW%=2:COL%=1:MAT=0:GOSUB 31110:IF FERR$<>"OK" THEN 1290
1180 ROW%=1:COL%=KDIM(1)+1:MAT=1:GOSUB 31110:IF FERR$<>"OK" THEN 1290
1190 ROW%=1:COL%=KDIM(2)+1:MAT=2:GOSUB 31110:IF FERR$<>"OK" THEN 1290 ELSE EM=1
1200 CLOSE #FILENUM:COLOR 7,0:LOCATE 14,30,0:PRINT"SEE MODEL? (Y/N)"
1210 INFLG=1:GOSUB 7120:IF C$="Y" AND ESCK=0 THEN 1240
1220 IF ARC%=1 THEN COM$="Model recalled from file"+SFILE$:GOSUB 30410
1230 RETURN
1240 CLS:COLOR 12,0:LOCATE 10,31,0:PRINT"N(s) DIMENSION >":LOCATE 12,31,0:PRINT"D(s) DIMENSION >"
1250 COLOR 6,0:LOCATE 10,48,0:PRINT KDIM(1):LOCATE 12,48,0:PRINT KDIM(2)
1260 GOSUB 30850:IF ESCK=1 THEN 1220 ELSE P=1:NUMDIG=12:CLS
1270 YFLD=3:XFLD=1:MAT=1:COM$="N(s) POLYNOMIAL":ROW%=1:COL%=KDIM(1)+1:GOSUB 30890
1280 YFLD=9:XFLD=1:MAT=2:COM$="D(s) POLYNOMIAL":ROW%=1:COL%=KDIM(2)+1:GOSUB 30890:GOSUB 30850:GOTO 1220
1290 CLOSE #FILENUM:COM$=FERR$:GOSUB 32700:IF C$="Y" AND ESCK=0 THEN 1110 ELSE EM=0
1300 IF ARC%=1 THEN COM$=FERR$+"for file"+SFILE$:GOSUB 30410
1310 GOTO 1140
1320 '
1330 'INPUT PARAMETERS
1340 IF EM=0 THEN COM$="ENTER MODEL FIRST!!":GOSUB 32310:RETURN
1350 CLS:IP=0:IF HELP%=0 THEN 1430 ELSE COLOR 3,0
1360 LOCATE 2,12,0:PRINT"You will be asked to enter the gain, K, the minimum"
1370 LOCATE 3,12,0:PRINT"frequency, FMIN, the maximum frequency, FMAX, and the"
1380 LOCATE 4,12,0:PRINT"number of frequencies to be used, FNO. For Bode "
1390 LOCATE 5,12,0:PRINT"plots, frequency is interpolated logarithmically between"
1400 LOCATE 6,12,0:PRINT"FMIN and FMAX while for Nyquist and Nichols frequency"
1410 LOCATE 7,12,0:PRINT"is interpolated linearly. Frequency in radians/sec."

```

```

1415 LOCATE 8,12,0:PRINT"FNO must not exceed"MAXPTS
1420 GOSUB 30850:IF ESCK=1 THEN 1550 ELSE CLS
1430 COLOR 12,0:LOCATE 4,31,0:PRINT"SYSTEM GAIN  >"SPC(20)
1440 XFLD=48:YFLD=4:NUMDIG=12:INFLG=3:GOSUB 7110:IF ESCK=1 THEN 155
0 ELSE KDIM(4)=S
1450 IF KDIM(4)=0 THEN GOSUB 30760:COLOR 15,0:LOCATE 6,20,0:PRINT"E
RROR!  Value for gain must be nonzero.":GOTO 1430
1460 COLOR 12,0:LOCATE 8,31,0:PRINT"MINIMUM FREQ  >"SPC(20)
1470 XFLD=48:YFLD=8:NUMDIG=12:INFLG=3:GOSUB 7110:IF ESCK=1 THEN 155
0
1475 IF S>0 THEN KDIM(5)=S:GOTO 1490
1480 GOSUB 30760:COLOR 15,0:LOCATE 10,13,0:PRINT"ERROR!  Minimum fr
equency must be positive and nonzero.":GOTO 1460
1490 COLOR 12,0:LOCATE 12,31,0:PRINT"MAXIMUM FREQ  >"SPC(20)
1500 XFLD=48:YFLD=12:NUMDIG=12:INFLG=3:GOSUB 7110:IF ESCK=1 THEN 15
50
1502 IF S>=KDIM(5) THEN KDIM(6)=S:GOTO 1504
1503 GOSUB 30760:COLOR 15,0:LOCATE 14,7,0:PRINT"ERROR!  Maximum fre
quency must be greater than minimum frequency.":GOTO 1490
1504 COLOR 12,0:LOCATE 16,31,0:PRINT"NUMBER FREQS  >"SPC(20)
1506 XFLD=48:YFLD=16:NUMDIG=12:INFLG=3:GOSUB 7110:IF ESCK=1 THEN 15
50
1508 S=INT(S):IF S<=MAXPTS AND S>0 THEN KDIM(3)=S:GOTO 1530
1510 GOSUB 30760:COLOR 15,0:LOCATE 18,18,0:PRINT"ERROR!  Value must
lie in the range 1 to"MAXPTS:GOTO 1504
1530 IF ARC%=1 THEN GOSUB 31710:COM$="-INPUT PARAMETERS-":GOSUB 304
10
1540 IP=1:RETURN
1550 IP=0:COM$="INPUTS NOT ENTERED!!":GOSUB 32310:RETURN
1560 '
1690 'MODIFY CURRENT MODEL
1700 IF EM=0 THEN COM$="NO MODEL TO MODIFY!!":GOSUB 32310:RETURN
1710 IF ARC%=1 THEN GOSUB 31710:COM$="-MODIFY CURRENT MODEL-":GOSUB
30410
1720 CLS:IF HELP%=0 THEN 1740 ELSE LOCATE 1,14,0:COLOR 3,0
1730 PRINT"ENTER LETTER OF POLYNOMIAL TO CHANGE, ESC TO QUIT"
1740 COLOR 12,0:LOCATE 3,30,0:PRINT"(N, D, OR ESC) >"SPC(5):INFLG=5
:NUMDIG=1:YFLD=3:XFLD=49
1750 GOSUB 7110:IF ESCK=1 THEN RETURN
1770 IF S$="N" OR S$="n" THEN COM$="N(s) POLYNOMIAL":MAT=1:ROW%=1:C
OL%=KDIM(1):IF COL%=0 THEN 1790 ELSE GOSUB 32440:GOTO 1720
1780 IF S$="D" OR S$="d" THEN COM$="D(S) POLYNOMIAL":MAT=2:ROW%=1:C
OL%=KDIM(2):IF COL%=0 THEN 1790 ELSE GOSUB 32440:GOTO 1720
1790 GOSUB 30760:GOTO 1740
1800 '
1810 'CALCULATE DATA
1820 IF EM=0 THEN COM$="ENTER MODEL FIRST!!":GOSUB 32310:RETURN
1830 IF IP=0 THEN COM$="ENTER INPUTS FIRST!!":GOSUB 32310:RETURN
1840 CLS:COLOR 12,0:LOCATE 9,23,0:PRINT"<Sft-F7>....BODE CALCULATIO
NS"
1850 LOCATE 10,23,0:PRINT"<Sft-F8>....NYQUIST CALCULATIONS"

```

```

1855 LOCATE 11,23,0:PRINT"<Sft-F9>....NICHOLS CALCULATIONS"
1860 INFLG=2:GOSUB 7120:IF ESCK=1 THEN RETURN
1870 IF C=90 THEN KDIM(7)=0:GOTO 1900
1880 IF C=91 THEN KDIM(7)=1:GOTO 1900
1885 IF C=92 THEN KDIM(7)=2:GOTO 1900
1890 GOSUB 30760:GOTO 1860
1900 CLS:LOCATE 11,29,0:COLOR 3,0:PRINT"CALCULATING DATA.....":GA
IN=0
1905 IF ARC%=1 THEN GOSUB 31710:COM$="-CALCULATED DATA-":GOSUB 3041
0
1906 FOR I=1 TO KDIM(1)+1:C(1,I)=A(I):NEXT I
1910 OP(1,1)=KDIM(5):IF KDIM(3)=1 THEN 1960
1920 IF KDIM(7)=0 THEN 1940 ELSE WINT=(KDIM(6)-KDIM(5))/KDIM(3)
1930 FOR I=2 TO KDIM(3)+1:OP(I,1)=OP(I-1,1)+WINT:NEXT I:GOTO 1960
1940 WINT=(LOG(KDIM(6))/LOG(10)-LOG(KDIM(5))/LOG(10))/KDIM(3):WINT=
10*WINT
1950 FOR I=2 TO KDIM(3)+1:OP(I,1)=OP(I-1,1)*WINT:NEXT I
1960 IF KDIM(7)<>1 THEN 2100
1962 FOR I=1 TO KDIM(1)+1:C(1,I)=KDIM(4)*C(1,I):NEXT I
1965 DEF FNCPLXR(R1,AI1,R2,AI2)=R1*R2-AI1*AI2
1970 DEF FNCPLXI(R1,AI1,R2,AI2)=R1*AI2+R2*AI1
1980 FOR K=1 TO KDIM(3)+1:SR=0:SI=OP(K,1):PR=C(1,KDIM(1)+1):PPI=0
1985 IF KDIM(1)=0 THEN QNUMR=C(1,1):QNUMI=0:GOTO 2020
1990 FOR J=1 TO KDIM(1):QNUMR=FNCPLXR(PR,PPI,SR,SI):QNUMR=QNUMR+C(1
,KDIM(1)+1-J)
2010 QNUMI=FNCPLXI(PR,PPI,SR,SI):PR=QNUMR:PPI=QNUMI:NEXT J
2020 PR=B(KDIM(2)+1):PPI=0:IF KDIM(2)=0 THEN QDENR=B(1):QDENI=0:GOT
O 2050
2030 FOR J=1 TO KDIM(2):QDENR=FNCPLXR(PR,PPI,SR,SI):QDENR=QDENR+B(K
DIM(2)+1-J)
2040 QDENI=FNCPLXI(PR,PPI,SR,SI):PR=QDENR:PPI=QDENI:NEXT J
2050 IF SQR(QDENI^2+QDENR^2)=0 THEN OP(K,2)=1E+30:OP(K,3)=1E+30:GOT
O 2090
2070 OP(K,2)=(QNUMR*QDENR+QNUMI*QDENI)/(QDENR^2+QDENI^2)
2080 OP(K,3)=(QNUMI*QDENR-QNUMR*QDENI)/(QDENR^2+QDENI^2)
2090 NEXT K:RS=1:RETURN
2100 N=KDIM(1):FOR I=1 TO KDIM(1)+1:SDI(I)=C(1,I):NEXT I:GOSUB 3279
0
2110 FOR I=1 TO KDIM(1):C(1,I)=-SP0(I,1):C(2,I)=-SP0(I,2):NEXT I
2120 N=KDIM(2):FOR I=1 TO KDIM(2)+1:SDI(I)=B(I):NEXT I:GOSUB 32790
2130 FOR I=1 TO KDIM(2):D(1,I)=-SP0(I,1):D(2,I)=-SP0(I,2):NEXT I
2140 N=1:FOR K=1 TO KDIM(3):OP(K,2)=0:OP(K,3)=0
2145 OP(K,2)=20*FNLOG10(KDIM(4))
2150 FOR J=1 TO KDIM(1):OP(K,2)=OP(K,2)+20*FNLOG10(SQR((C(2,J)+OP(K
,1))^2+C(1,J)^2))
2160 SR=C(1,J):SI=C(2,J)+OP(K,1):GOSUB 34610:OP(K,3)=OP(K,3)+ANGLE:
NEXT J
2170 FOR J=1 TO KDIM(2):OP(K,2)=OP(K,2)-20*FNLOG10(SQR((D(2,J)+OP(K
,1))^2+D(1,J)^2))
2180 SR=D(1,J):SI=D(2,J)+OP(K,1):GOSUB 34610:OP(K,3)=OP(K,3)-ANGLE:
NEXT J

```

```

2190 OP(K,3)=OP(K,3)*180/3. 141593:NEXT K: RS=1: RETURN
2800 '
2810 ' GRAPHING ROUTINE
2820 RG=0: IF ARC%=1 THEN GOSUB 31710: COM$="-GRAPH DATA-": GOSUB 3041
0
2830 SCREEN 0,1: CLS: COLOR 14,0: LOCATE 3,36,0: PRINT"OPTIONS": COLOR 1
0,0
2840 LOCATE 5,24,0: PRINT"<Ctrl-F1>....GRAPH CURRENT DATA"
2850 LOCATE 6,24,0: PRINT"<Ctrl-F2>....RE-DISPLAY SCREEN FILE"
2860 LOCATE 7,24,0: PRINT"<ESC>.....END"
2870 COLOR 14,0: LOCATE 12,29,0: PRINT"WHILE GRAPH DISPLAYED": COLOR 1
0,0
2880 LOCATE 14,24,0: PRINT"<ENTER>.....CURSOR COORDINATES"
2900 LOCATE 15,24,0: PRINT"<Ctrl-F4>....SAVE GRAPH ON DISK"
2910 LOCATE 16,24,0: PRINT"<Ctrl-F5>....ZOOM"
2915 LOCATE 17,24,0: PRINT"<Ctrl-F10>...NICHOLS TRANSLATION"
2920 LOCATE 18,24,0: PRINT"<ESC>.....END": COLOR 14,0
2930 LOCATE 20,29,0: PRINT"(EXCEPT ON RE-DISPLAY)"
2934 IF KDIM(7)=2 AND GAIN<>0 THEN COLOR 28,0: LOCATE 25,44,0: PRINT"
NICHOLS MAGN MODIFIED BY TRANSLATION";
2940 INFLG=2: GOSUB 7120: IF ESCK=1 THEN RETURN
2950 IF C=94 THEN GOSUB 3000: GOTO 2830
2960 IF C=95 THEN GOSUB 3400: GOTO 2830
2970 GOSUB 30760: GOTO 2940
2980 '
2990 ' GRAPH CURRENT DATA
3000 IF RS=0 THEN COM$="NO DATA TO GRAPH!": GOSUB 32310: RETURN
3010 CLS: COLOR 12,0: IF ARC%=1 THEN COM$="Graphed current data": GOSU
B 30410
3013 PN=KDIM(3)+1: IF KDIM(7)=0 THEN 3080
3015 LOCATE 11,26,0: PRINT"GRIDS ON GRAPH? (Y/N)"
3020 IF HELP%=1 THEN LOCATE 12,28,0: PRINT"(Default = No)"
3025 INFLG=1: GOSUB 7110: IF C$="N" THEN PGD=0 ELSE PGD=1
3030 IF KDIM(7)=1 THEN FOR I=1 TO KDIM(3): PX(I)=OP(I,2): PY(I)=OP(I,
3): NEXT I: GOTO 3050
3035 FOR I=1 TO KDIM(3): PX(I)=OP(I,3): PY(I)=OP(I,2): NEXT I
3050 PFM=0: PFLG=0: GOSUB 4620: IF P=0 THEN RETURN
3060 GOSUB 6020: IF ZOOM=0 AND TRANS=0 THEN RETURN ELSE PFM=1: PFLG=1
: GOSUB 4620
3070 IF P=0 THEN RETURN ELSE PFLG=2: GOSUB 4620: GOTO 3060
3080 LOCATE 10,19,0: PRINT"PLOT MAGN. AND PHASE SEPARATELY? (Y/N)"
3090 IF HELP%=1 THEN LOCATE 11,31,0: PRINT"(Default = No)"
3100 INFLG=1: GOSUB 7120: IF C$="N" THEN PCT=1 ELSE PCT=0
3120 PMINX=FNLOG10(KDIM(5)): PMAXX=FNLOG10(KDIM(6))
3125 FOR I=1 TO KDIM(3): PX(I)=FNLOG10(OP(I,1)): NEXT I
3130 IF PCT=1 THEN 3135 ELSE PP=0: GOSUB 3160
3132 IF P<>0 THEN GOSUB 6020: IF ZOOM<>0 THEN GOSUB 8030: GOTO 3132
3133 PMINX=FNLOG10(KDIM(5)): PMAXX=FNLOG10(KDIM(6)): PP=1: GOSUB 3210
3134 IF P=0 THEN RETURN ELSE GOSUB 6020: IF ZOOM=0 THEN RETURN ELSE
GOSUB 8030: GOTO 3134
3135 PP=0: GOSUB 3210: IF P<>0 THEN 3144 ELSE PCT=0: PP=0: GOSUB 3160

```

```

3136 IF P=0 THEN RETURN ELSE GOSUB 6020: IF ZOOM=0 THEN RETURN ELSE
GOSUB 8030: GOTO 3136
3144 PP=1: GOSUB 3160
3145 INFLG=2: GOSUB 7120: IF ESCK=1 THEN RETURN
3146 IF C=97 THEN GOSUB 6670: GOTO 3145
3148 GOSUB 30760: GOTO 3145
3150 'GRAPH MAGNITUDE
3160 PMAXY=0: PMINY=0: FOR I=1 TO KDIM(3): PY(I)=OP(I,2)
3170 IF PMAXY<PY(I) THEN PMAXY=PY(I)
3180 IF PMINY>PY(I) THEN PMINY=PY(I)
3190 NEXT I: GOSUB 8030: RETURN
3200 'GRAPH PHASE
3210 PMINY=0: PMAXY=0: FOR I=1 TO KDIM(3): PY(I)=OP(I,3)
3220 IF PMAXY<PY(I) THEN PMAXY=PY(I)
3230 IF PMINY>PY(I) THEN PMINY=PY(I)
3240 NEXT I: GOSUB 8030: RETURN
3380 '
3390 'RE-GRAPH SCREEN FILE
3400 CLS: FILETPE=1: IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
3410 FILEXT$=".GPH": FILENUM=1: GOSUB 30650
3420 IF ESCK=1 OR FERR$<>"OK" THEN RETURN ELSE CLOSE #FILENUM: RG=1
3430 CLS: SCREEN 2: DEF SEG=&HB800
3440 ON ERROR GOTO 3510: BLOAD FILENAM$,0: DEF SEG: ON ERROR GOTO 0
3450 IF ARC%=1 THEN COM$="Re-graph from screen file "+FILENAM$: GOSU
B 30410
3470 INFLG=2: GOSUB 7120: IF ESCK=1 THEN RETURN
3500 GOSUB 30760: GOTO 3470
3510 ON ERROR GOTO 0: CLS: SCREEN 0,1: COLOR 15,0: GOSUB 30760: LOCATE 1
0,30,0
3520 PRINT"ERROR GRAPHING FILE": COLOR 7,0: LOCATE 12,33,0: PRINT RT$:
INFLG=1
3530 GOSUB 7120: IF C$="Y" AND ESCK=0 THEN RESUME 3400 ELSE RESUME 3
540
3540 IF ARC%=1 THEN COM$="ERROR REGRAPHING FILE "+FILENAM$: GOSUB 30
410
3560 RETURN
3570 '
3580 'SAVE/LIST DATA
3590 IF ARC%=1 THEN GOSUB 31710: COM$="-SAVE/LIST DATA-": GOSUB 30410
3600 CLS: COLOR 14,0: LOCATE 4,36,0: PRINT"OPTIONS": COLOR 10,0
3610 LOCATE 6,25,0: PRINT"<Ctrl-F6>....LIST CURRENT DATA"
3620 LOCATE 7,25,0: PRINT"<Ctrl-F7>....SAVE DATA ON DISK"
3630 LOCATE 8,25,0: PRINT"<Ctrl-F8>....RECALL DATA FILE"
3640 LOCATE 9,25,0: PRINT"<ESC>.....END"
3645 IF KDIM(7)=2 AND GAIN<>0 THEN COLOR 28,0: LOCATE 25,44,0: PRINT"
NICHOLS MAGN MODIFIED BY TRANSLATION";
3650 INFLG=2: GOSUB 7120: IF ESCK=1 THEN RETURN
3660 IF C=99 THEN GOSUB 3720: GOTO 3600
3670 IF C=100 THEN GOSUB 3890: GOTO 3600
3680 IF C=101 THEN GOSUB 4070: GOTO 3600
3690 GOSUB 30760: GOTO 3650

```

```

3700 '
3710 'LIST CURRENT DATA
3720 IF RS=0 THEN COM$="NO DATA TO LIST!!":GOSUB 32310:RETURN
3730 IF ARC%=1 THEN COM$="Listed current data":GOSUB 30410
3750 YFLD=3:XFLD=19:LCD=21:CLS:COLOR 6,0
3760 FOR I=1 TO KDIM(3):IF I/LCD<1 THEN 3780 ELSE LCD=LCD+21:GOSUB
3810
3770 IF ESCK=1 THEN RETURN
3780 LOCATE YFLD,1,0:PRINT OP(I,1):LOCATE YFLD,XFLD,0:PRINT OP(I,2)
;
3790 XFLD=XFLD+22:LOCATE YFLD,XFLD,0:PRINT OP(I,3);
3800 YFLD=YFLD+1:XFLD=19:NEXT I:GOSUB 3810:RETURN
3810 COLOR 12,0:LOCATE 1,1,0:PRINT"FREQ(rad/sec)";
3820 IF KDIM(7)=1 THEN PRINT SPC(10)"REAL"SPC(16)"IMAGINARY";:GOTO
3840
3830 PRINT SPC(5)"MAGNITUDE(db)"SPC(13)"PHASE(deg)";
3840 XFLD=28:YFLD=25:GOSUB 30800:CLS:XFLD=19:YFLD=3:COLOR 6,0:RETUR
N
3870 '
3880 'SAVE DATA ON DISK
3890 IF RS=0 THEN COM$="NO DATA TO SAVE!!":GOSUB 32310:RETURN
3910 CLS:FILETPE=0:IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
3920 FILEXT$=".FDT":FILENUM=1:GOSUB 30650:IF ESCK=0 AND FERR$="OK"
THEN 3960
3940 COM$="DATA NOT SAVED!!":GOSUB 32310:RETURN
3960 SFILE$=FILENAM$:CLS:COLOR 3,0:LOCATE 11,31,0:PRINT"SAVING DATA
....."
3970 MAT=0:ROW%=7:COL%=1:GOSUB 31380:IF FERR$<>"OK" THEN 4010
3980 MAT=6:ROW%=KDIM(3):COL%=3:GOSUB 31380:IF FERR$<>"OK" THEN 4010
ELSE CLOSE #FILENUM
3990 IF ARC%=1 THEN COM$="Data saved in file "+SFILE$:GOSUB 30410
4000 RETURN
4010 CLOSE #FILENUM:COM$=FERR$:GOSUB 32700:IF C$="Y" AND ESCK=0 THE
N 3910
4030 IF ARC%=1 THEN COM$=FERR$+" while saving data":GOSUB 30410
4040 FILENAM$=SFILE$:GOSUB 31530:GOTO 3940
4050 '
4060 'RECALL DATA FILE
4070 IF RS=1 THEN COM$="ERASE EXISTING DATA? (Y/N)":GOSUB 32340:IF
C$="N" OR ESCK=1 THEN RETURN
4100 CLS:FILETPE=1:IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
4110 FILEXT$=".FDT":FILENUM=1:GOSUB 30650
4120 IF ESCK=0 AND FERR$="OK" THEN SFILE$=FILENAM$:GOTO 4160
4130 COM$="DATA NOT RECALLED!!":GOSUB 32310:RETURN
4160 CLS:COLOR 3,0:LOCATE 10,29,0:PRINT"RECALLING DATA....."
4165 TEMP1=KDIM(4):TEMP2=KDIM(5):TEMP3=KDIM(6)
4170 ROW%=7:COL%=1:MAT=0:GOSUB 31110:IF FERR$<>"OK" THEN 4210
4175 KDIM(4)=TEMP1:KDIM(5)=TEMP2:KDIM(6)=TEMP3
4180 ROW%=KDIM(3):COL%=3:MAT=6:GOSUB 31110:IF FERR$<>"OK" THEN 4210
4190 CLOSE #FILENUM:IF ARC%=1 THEN COM$="Recalled data from file "+
SFILE$:GOSUB 30410

```

```

4200 RS=1: GAIN=0: RETURN
4210 CLOSE #FILENAME: COM$=FERR$: GOSUB 32700: IF C$="Y" AND ESCK=0 THE
N 4100 ELSE RS=0: GOTO 4130
4570 '
4580 ' PLOTTING SUBROUTINE
4590 ' PFLG=0 -axes & plot : PFLG=1 -axes only : PLFG=2 -plot only
4600 ' PGD=1 grids on plot : PGD=0 no grids : PFM=1 don't find min/m
ax
4610 ' Plots PX() and PY() arrays where PN = number of xy pairs
4620 IF PFLG>1 THEN GOSUB 5490: GOTO 4670
4630 GOSUB 4720: GOSUB 6820
4640 IF PFM=1 THEN GOSUB 5170: IF P=0 THEN RETURN ELSE GOSUB 4680: GO
SUB 5240: GOTO 4660
4650 GOSUB 4820: GOSUB 5170: IF P=0 THEN RETURN ELSE GOSUB 4680: GOSUB
5240
4660 IF PFLG=1 THEN RETURN ELSE GOSUB 5450
4670 IF KDIM(7)<2 THEN RETURN ELSE GOSUB 5974: RETURN
4680 ' TITLE
4690 LOCATE 1,7
4700 PRINT "SCALE X-AXIS BY"; STR$(PSCLX); ", Y-AXIS BY"; PSCLY;
4710 RETURN
4720 ' INITIALIZATION
4750 PNINX=6.5: SCREEN 2: CLS: PN=PN-1: IF PCT=0 THEN PNINY=5.5 ELSE PN
INY=4.5
4760 PLNL=50: PLNR=589: PLNB=173: PLNT=12: PLOCX1=8
4770 PLOCY1=INT(PLNT/8+.5): PLOCY2=PLOCY1+1
4780 PLOCY3=INT(PLNB/8+.625): PLOCY4=PLOCY3+1
4790 PLOCY5=PLOCY4+1: PLOCX2=INT(PLNL/8-4)
4800 PLOCX3=PLOCX2+1: PLOCX4=PLOCX3+INT((PLNR-PLNL)/8+.625)-1
4810 RETURN
4820 ' FIND MINIMUM & MAXIMUM OF ARRAYS
4830 PMINX=PX(1): PMAXX=PMINX: PMINY=PY(1): PMAXY=PMINY
4840 FOR PII%=2 TO PN
4850 IF PMINX>PX(PII%) THEN PMINX=PX(PII%)
4860 IF PMAXX<PX(PII%) THEN PMAXX=PX(PII%)
4870 IF PMINY>PY(PII%) THEN PMINY=PY(PII%)
4880 IF PMAXY<PY(PII%) THEN PMAXY=PY(PII%)
4890 NEXT PII%
4900 RETURN
4910 ' FIND LINEAR SCALE INTERVALS
4920 IF PMIN=PMAX THEN PMIN=PMIN-1: PMAX=PMAX+1
4950 P=1: PEPS=.025: PA=ABS(PMIN)
4960 IF ABS(PMIN)<ABS(PMAX) THEN PA=ABS(PMAX)
4970 PSCL=10*INT(LOG(PA)/LOG(10))
4980 PMINA=PMIN/PSCL: PMAXA=PMAX/PSCL
4990 PD=(PMAXA-PMINA)/PNIN: PJ=PD*PEPS: PE=INT(LOG(PD)/LOG(10))
5000 PF=PD/10*PE: PV=10
5010 IF PF<SQR(50) THEN PV=5
5020 IF PF<SQR(10) THEN PV=2
5030 IF PF<SQR(2) THEN PV=1
5040 PWDT=PV*10*PE: PG=INT(PMINA/PWDT)

```

```

5050 IF ABS(PG+1-PMINA/PWDT)<PJ THEN PG=PG+1
5060 PMINP=PWDT*PG: PH=INT(PMAXA/PWDT)+1
5070 IF ABS(PMAXA/PWDT+1-PH)<PJ THEN PH=PH-1
5080 PMAXP=PWDT*PH: PNINP=PH-PG
5090 IF ABS(PMAXP)>=10 OR ABS(PMINP)>=10 THEN PSCL=PSCL*10: GOTO 498
0
5100 RETURN
5110 'LINEAR TRANSFORMATION FROM SUBJECT TO OBJECT SPACE
5120 PX1=FNPSC(PX(P11%-1)/PSCLX,PMAXPX,PMINPX,PLNR,PLNL)
5130 PX2=FNPSC(PX(P11%)/PSCLX,PMAXPX,PMINPX,PLNR,PLNL)
5140 PY1=FNPSC(PY(P11%-1)/PSCLY,PMAXPY,PMINPY,PLNT,PLNB)
5150 PY2=FNPSC(PY(P11%)/PSCLY,PMAXPY,PMINPY,PLNT,PLNB)
5160 RETURN
5170 'FIND X & Y LINEAR SCALE INTERVALS
5180 'X SCALE
5190 PMIN=PMINX: PMAX=PMAXX: PNIN=PNINX: GOSUB 4910: IF P=0 THEN RETURN
5200 PSCLX=PSCL: PMINPX=PMINP: PMAXPX=PMAXP: PNINPX=PNINP: PWDTX=PWDT
5210 'Y SCALE
5220 PMIN=PMINY: PMAX=PMAXY: PNIN=PNINY: GOSUB 4910: IF P=0 THEN RETURN
5230 PSCLY=PSCL: PMINPY=PMINP: PMAXPY=PMAXP: PNINPY=PNINP: PWDY=PWDT: R
ETURN
5240 'DRAW LINEAR GRIDS & LABELS
5250 PLOCWY=(PLOCY3-PLOCY2+1)/PNINPY: PLNWY=(PLNB-PLNT)/PNINPY: X1=PL
NL-49 : X2=PLNL-4
5260 'Y GRID & LABELS
5270 FOR P11=0 TO PNINPY: PLNY1=PLNB-P11*PLNWY
5280 Y1=PLNY1-5: Y2=PLNY1+5: S$=STR$(PMINPY+P11*PWDY): GOSUB 6850
5290 IF PGD=1 THEN 5320
5300 IF P11=0 OR P11=PNINPY THEN 5320
5310 LINE(PLNL-4,PLNY1)-(PLNL,PLNY1): LINE(PLNR,PLNY1)-(PLNR+4,PLNY1
): GOTO 5330
5320 LINE(PLNL,PLNY1)-(PLNR,PLNY1)
5330 NEXT P11
5340 PLOCWX=(PLOCX4-PLOCX3+1)/PNINPX: PLNWX=(PLNR-PLNL)/PNINPX
5350 'X GRID & LABELS
5360 FOR P11=0 TO PNINPX
5370 IF P11/2=INT(P11/2) THEN PLOC=PLOCY5 ELSE PLOC=PLOCY4
5380 PLNX1=PLNL+P11*PLNWX: Y1=PLOC*8-8: Y2=Y1+8: X1=PLNX1-30: X2=PLNX1+
25
5390 S$=STR$(PMINPX+P11*PWDX): GOSUB 6850: IF PGD=1 THEN 5420
5400 IF P11=0 OR P11=PNINPX THEN 5420
5410 LINE(PLNX1,PLNB+2)-(PLNX1,PLNB): LINE(PLNX1,PLNT)-(PLNX1,PLNT-2
): GOTO 5430
5420 LINE(PLNX1,PLNB)-(PLNX1,PLNT)
5430 NEXT P11
5440 RETURN
5450 'DRAW X-Y CURVE
5460 FOR P11%=2 TO PN: GOSUB 5110
5470 IF P11%>2 THEN LINE -(PX2,PY2) ELSE LINE(PX1,PY1)-(PX2,PY2)
5480 NEXT P11%: RETURN
5490 'CLIPPING ROUTINE TO CHECK FOR POINTS OUTSIDE GRID

```

```

5500 FOR PII%=2 TO PN:GOSUB 5110
5510 IF PX1<PLNL OR PX1>PLNR OR PY1>PLNB OR PY1<PLNT THEN PFL1=1 EL
SE PFL1=0
5520 IF PX2<PLNL OR PX2>PLNR OR PY2>PLNB OR PY2<PLNT THEN PFL2=1 EL
SE PFL2=0
5530 IF PFL1=0 AND PFL2=0 THEN 5960
5540 IF PFL1*PFL2=1 THEN 5760
5550 'ONE POINT OUTSIDE, ONE POINT INSIDE
5560 IF PX1=PX2 THEN 5660
5570 IF PFL1=1 AND PX1>PLNR OR PFL2=1 AND PX2>PLNR THEN 5620
5580 PYT=FNPS(PLNL,PX1,PX2,PY1,PY2)
5590 IF PYT>PLNB OR PYT<PLNT THEN 5620
5600 IF PFL2=1 THEN PX2=PLNL: PY2=PYT: GOTO 5960
5610 PX1=PLNL: PY1=PYT: GOTO 5960
5620 PYT=FNPS(PLNR,PX1,PX2,PY1,PY2)
5630 IF PYT>PLNB OR PYT<PLNT THEN 5660
5640 IF PFL2=1 THEN PX2=PLNR: PY2=PYT: GOTO 5960
5650 PX1=PLNR: PY1=PYT: GOTO 5960
5660 IF PY1=PY2 THEN 5760
5670 IF PFL1=1 AND PY1<PLNT OR PFL2=1 AND PY2<PLNT THEN 5720
5680 PXT=FNPS(PLNB,PY1,PY2,PX1,PX2)
5690 IF PXT<PLNL OR PXT>PLNR THEN 5720
5700 IF PFL2=1 THEN PY2=PLNB: PX2=PXT: GOTO 5960
5710 PY1=PLNB: PX1=PXT: GOTO 5960
5720 PXT=FNPS(PLNT,PY1,PY2,PX1,PX2)
5730 IF PFL2=1 THEN PY2=PLNT: PX2=PXT: GOTO 5960
5740 PY1=PLNT: PX1=PXT: GOTO 5960
5750 'BOTH POINTS LIE OUTSIDE
5760 PTEMP=0: IF PX1<PLNL AND PX2<PLNL OR PX1>PLNR AND PX2>PLNR THEN
5970
5770 IF PY1<PLNT AND PY2<PLNT OR PY1>PLNB AND PY2>PLNB THEN 5970
5780 IF PX1=PX2 THEN 5870
5790 PYT=FNPS(PLNL,PX1,PX2,PY1,PY2)
5800 IF PYT>PLNB OR PYT<PLNT THEN 5820
5810 PX1=PLNL: PY1=PYT: PTEMP=1
5820 PYT=FNPS(PLNR,PX1,PX2,PY1,PY2)
5830 IF PYT>PLNB OR PYT<PLNT THEN 5870
5840 PTEMP=PTEMP+1
5850 IF PTEMP=2 THEN PX2=PLNR: PY2=PYT: GOTO 5960
5860 PX1=PLNR: PY1=PYT
5870 IF PY1=PY2 THEN 5970
5880 PXT=FNPS(PLNB,PY1,PY2,PX1,PX2)
5890 IF PXT<PLNL OR PXT>PLNR THEN 5930
5900 PTEMP=PTEMP+1
5910 IF PTEMP=2 THEN PY2=PLNB: PX2=PXT: GOTO 5960
5920 PY1=PLNB: PX1=PXT
5930 IF PTEMP=0 THEN 5970
5940 PXT=FNPS(PLNT,PY1,PY2,PX1,PX2)
5950 PY2=PLNT: PX2=PXT
5960 LINE(PX1,PY1)-(PX2,PY2)
5970 NEXT PII%

```

```

5971 RETURN
5972 '
5973 'DRAW -3db LINE
5974 NDB=0: DBI=(PMAXX-PMINX)/100: THETA=PMINX
5976 FOR I=1 TO 100: GG=-2.00952*COS(THETA*.0174533): GGG=SQR(GG^2+4.
019041)
5978 G1=(-GG+GGG)/2: G2=(-GG-GGG)/2
5980 IF G1>0 THEN G=20*FNLOG10(G1): GOTO 5983
5982 IF G2>0 THEN G=20*FNLOG10(G2) ELSE 5984
5983 IF G>=PMINX AND G<=PMAXX THEN NDB=NDB+1: DB(NDB,1)=THETA: DB(NDB
,2)=G
5984 THETA=THETA+DBI: NEXT I: FOR I=2 TO NDB
5986 PX1=FNPSX(DB(I-1,1)/PSCLX,PMAXPX,PMINPX,PLNR,PLNL)
5988 PX2=FNPSX(DB(I,1)/PSCLX,PMAXPX,PMINPX,PLNR,PLNL)
5990 PY1=FNPSX(DB(I-1,2)/PSCLY,PMAXPY,PMINPY,PLNT,PLNB)
5992 PY2=FNPSX(DB(I,2)/PSCLY,PMAXPY,PMINPY,PLNT,PLNB)
5994 IF I>2 THEN LINE -(PX2,PY2) ELSE LINE (PX1,PY1)-(PX2,PY2)
5995 IF I=INT(NDB/2) THEN S$="-3db": X1=PX1-20: X2=PX1+20: Y1=PY1-12: Y
2=PY1: GOSUB 6900
5996 NEXT I: PY1=FNPSX(0/PSCLY,PMAXPY,PMINPY,PLNT,PLNB): PY2=PY1: PX1=
PLNL: PX2=PLNR
5997 IF PY1>=PLNT AND PY1<=PLNB THEN LINE (PX1,PY1)-(PX2,PY2)
5998 PX1=FNPSX(-180/PSCLX,PMAXPX,PMINPX,PLNR,PLNL): PX2=PX1: PY1=PLNB
: PY2=PLNT
5999 IF PX1>PLNR OR PX1<PLNL THEN RETURN ELSE LINE (PX1,PY1)-(PX2,P
Y2): RETURN
6000 '
6001 'CURSOR SUBROUTINE
6010 'Moves cursor around screen keeping pixel coordinates
6020 DEF SEG=0: POKE 1047,64: DEF SEG: IF SND%>1 THEN BEEP
6030 CRSRX=PLNL: CRSRY=PLNB: ZOOM=0: TRANS=0
6040 IF CRSRX<PLNL THEN CRSRX=PLNL
6050 IF CRSRX>PLNR THEN CRSRX=PLNR
6060 IF CRSRY<PLNT THEN CRSRY=PLNT
6070 IF CRSRY>PLNB THEN CRSRY=PLNB
6080 GOSUB 6300: GOSUB 30030 'Draw csr & clear KBD buf
6090 C$=INKEY$: IF C$="" THEN 6090
6100 IF C$=CHR$(27) THEN ZOOM=0: RETURN
6110 IF C$=CHR$(13) THEN GOSUB 6470 'Wants coordinates
6120 IF LEN(C$)=2 THEN C$=RIGHT$(C$,1) ELSE 6230
6150 IF C$=CHR$(98) THEN GOSUB 6550: IF ESCK<>1 THEN ZOOM=1: RETURN
6155 IF C$=CHR$(103) AND KDIM(7)=2 THEN GOSUB 6643: IF ESCK<>1 THEN
TRANS=1: RETURN
6160 IF C$=CHR$(97) THEN GOSUB 6670
6170 GOSUB 6380 'Erase CRSR
6180 IF C$=CHR$(72) THEN CRSRY=CRSRY-1: GOTO 6040
6190 IF C$=CHR$(80) THEN CRSRY=CRSRY+1: GOTO 6040
6200 IF C$=CHR$(77) THEN CRSRX=CRSRX+1: GOTO 6040
6210 IF C$=CHR$(75) THEN CRSRX=CRSRX-1: GOTO 6040
6220 DEF SEG=0: POKE 1047,64: DEF SEG: GOTO 6080
6230 GOSUB 6380 'Erase CRSR

```

```

6240 IF C$=CHR$(56) THEN CRSRY=CRSRY-5: GOTO 6040
6250 IF C$=CHR$(50) THEN CRSRY=CRSRY+5: GOTO 6040
6260 IF C$=CHR$(54) THEN CRSRX=CRSRX+10: GOTO 6040
6270 IF C$=CHR$(52) THEN CRSRX=CRSRX-10: GOTO 6040
6280 DEF SEG=0: POKE 1047,64: DEF SEG: GOTO 6080
6290 'DRAW CURSOR
6300 CRSRPX=CRSRX-6: CRSRPY=CRSRY-6
6310 FOR CII%=1 TO 11
6320 CRSRPX=CRSRPX+1: CRSRPY=CRSRPY+1
6330 SW(CII%,1)=POINT(CRSRPX,CRSRY): SW(CII%,2)=POINT(CRSRX,CRSRPY)
6340 PSET(CRSRPX,CRSRY): PSET(CRSRX,CRSRPY)
6350 NEXT CII%
6360 RETURN
6370 'ERASE CURSOR
6380 CRSREX=CRSRX-6: CRSREY=CRSRY-6
6390 FOR CII%=1 TO 11
6400 CRSREX=CRSREX+1: CRSREY=CRSREY+1
6410 PRESET(CRSREX,CRSRY): PRESET(CRSRX,CRSREY)
6420 IF SW(CII%,1)=1 THEN PSET(CRSREX,CRSRY)
6430 IF SW(CII%,2)=1 THEN PSET(CRSRX,CRSREY)
6440 NEXT CII%
6450 RETURN
6460 'COORDINATE FINDER
6470 PBYPY=PMINPY: PTYPY=PMINPY+PNINPY*PWDY
6475 IF KDIM(7)=0 THEN PXCOD=10*((CRSRX-PLNL)/PPDEC-ABS(PMINX)): GOT
O 6500
6480 PLX=PMINPX: PRX=PMINPX+PNINPX*PWDY
6490 PXCOD=(PRX-(PLNR-CRSRX))*(PRX-PLX)/(PLNR-PLNL)*PSCLX
6500 PYCOD=(PBYPY-(PLNB-CRSRY))*(PBYPY-PTY)/(PLNB-PLNT)*PSCLY
6505 IF KDIM(7)=2 THEN GOSUB 33160: GOTO 6522
6510 LOCATE 1,47: PRINT USING "X= ###.###ccc "; PXCOD
6520 LOCATE 1,64: PRINT USING "Y= ###.###ccc"; PYCOD: RETURN
6522 LOCATE 1,47: PRINT USING "MOL= ###.###ccc "; PYCOD
6524 LOCATE 1,64: PRINT USING "POL= ###.###ccc"; PXCOD
6526 GOSUB 30030: GOSUB 30810: IF ESCK=1 THEN RETURN
6528 G=10*(PYCOD/20): ANGLE=PXCOD*.0174533: PYCOD=G/SQR(G^2+2*G*COS(ANG
LE)+1)
6530 PYCOD=20*FNLOG10(PYCOD): PXCOD=57.29578*ATN(SIN(ANGLE)/(G+COS(ANG
LE)))
6532 LOCATE 1,47: PRINT USING "MCL= ###.###ccc "; PYCOD
6534 LOCATE 1,64: PRINT USING "PCL= ###.###ccc"; PXCOD
6535 GOSUB 30030: GOSUB 30810: IF ESCK=1 THEN RETURN
6536 LOCATE 1,47,0: IF FF=0 THEN PRINT "FREQUENCY NOT AVAILABLE
": RETURN
6537 IF FF=1 THEN PRINT USING " FREQUENCY= ###.###ccc ";W:
RETURN
6538 PRINT USING "FREQ. ###.###ccc ";W1: LOCATE 1,63: PRINT USING "TO
###.###ccc";W2: RETURN
6539 '
6545 'ZOOM ROUTINE
6550 LOCATE 1,47,0: PRINT "X1=" SPC(30): XFLD=51: YFLD=1:

```

```

6560 NUMDIG=10: INFLG=3: GOSUB 7120: IF ESCK=1 THEN 6640 ELSE TPMINX=S
6570 LOCATE 1,62,0: PRINT "X2=" SPC(14): XFLD=66: YFLD=1: GOSUB 7120: IF E
SCK=1 THEN 6640 ELSE TPMAXX=S
6580 IF TPMAXX<=TPMINX THEN GOSUB 30760: GOTO 6570
6590 LOCATE 1,47,0: PRINT "Y1=" SPC(30): XFLD=51: YFLD=1
6600 GOSUB 7120: IF ESCK=1 THEN 6640 ELSE TPMINY=S
6610 LOCATE 1,62,0: PRINT "Y2=" SPC(14): XFLD=66: YFLD=1: GOSUB 7120: IF E
SCK=1 THEN 6640 ELSE TPMAXY=S
6620 IF TPMAXY<=TPMINY THEN GOSUB 30760: GOTO 6610
6625 PMINY=TPMINY: PMAXY=TPMAXY: IF KDIM(7)>0 THEN PMAXX=TPMAXX: PMINX
=TPMINX: RETURN
6630 PMINX=FNLOG10(TPMINX): PMAXX=FNLOG10(TPMAXX): RETURN
6640 LOCATE 1,47,0: PRINT SPC(30): RETURN
6641 '
6642 'NICHOLS TRANSLATION ROUTINE
6643 LOCATE 1,47,0: PRINT "GAIN(db)=" SPC(24): XFLD=57: YFLD=1: NUMDIG=10
: INFLG=3
6644 GOSUB 7120: IF ESCK=1 THEN 6646 ELSE GAIN=S: PMAXY=PMAXY+GAIN: PM
INY=PMINY+GAIN
6645 FOR I=1 TO KDIM(3)+1: PY(I)=PY(I)+GAIN: OP(I,2)=OP(I,2)+GAIN: NEX
T I: RETURN
6646 LOCATE 1,47,0: PRINT SPC(30): RETURN
6650 '
6660 'SCREEN DUMP TO DISK
6670 LOCATE 1,47,0: PRINT "FILENAME >" SPC(24): NUMDIG=10: YFLD=1: XFLD=5
8
6680 INFLG=4: GOSUB 7120: IF ESCK=1 THEN 6800 ELSE FILENUM=1: FILETPE=
0
6690 FILENAM$=S$+".GPH": GOSUB 30090
6700 IF FERR$="OK" THEN CLOSE #FILENUM: LOCATE 1,47,0: PRINT SPC(30):
GOTO 6750
6710 LOCATE 1,47,0: PRINT FERR$ SPC(1) RT$: GOSUB 30760: INFLG=1: GOSUB
7120
6720 IF C$="Y" AND ESCK=0 THEN 6670 ELSE LOCATE 1,47,0: PRINT SPC(30
)
6730 IF ARC%=1 THEN COM$=FERR$+" on screen dump to disk": GOSUB 3041
0
6740 GOSUB 31530: RETURN
6750 ON ERROR GOTO 6790: DEF SEG=&HB800: BSAVE FILENAM$,0,&H4000: DEF
SEG: ON ERROR GOTO 0
6770 IF ARC%=1 THEN COM$="Saved Graphics Screen in file "+FILENAM$:
GOSUB 30410
6780 RETURN
6790 ON ERROR GOTO 0: GOSUB 31260: RESUME 6710
6800 LOCATE 1,47,0: PRINT SPC(30): RETURN
6810 '
6818 'TITLE ON GRAPH
6820 IF KDIM(7)=1 THEN 6830
6822 LOCATE 25,37,0: PRINT "PHASE";: S$="MAGNITUDE": FOR I=1 TO 9: T$=LE
FT$(S$,1)
6824 S$=MID$(S$,2): LOCATE I+7,1,0: PRINT T$: NEXT I: RETURN

```

```

6830 LOCATE 25,38,0:PRINT"REAL";:S$="IMAGINARY":FOR I=1 TO 9:T$=LEF
T$(S$,1)
6832 S$=MID$(S$,2):LOCATE I+7,1,0:PRINT T$:NEXT I:RETURN
6834 LOCATE 25,36,0:PRINT"FREQUENCY";:IF PCT=1 THEN 6838
6835 IF PP=0 THEN S$="MAGNITUDE" ELSE S$=" PHASE "
6836 FOR I=1 TO 9:T$=LEFT$(S$,1):S$=MID$(S$,2):LOCATE I+7,1,0:PRINT
T$:NEXT I:RETURN
6838 S$="MAGNITUDE":FOR I=1 TO 9:T$=LEFT$(S$,1):S$=MID$(S$,2):LOCAT
E I+2,1,0:PRINT T$:NEXT I
6839 S$="PHASE":FOR I=1 TO 5:T$=LEFT$(S$,1):S$=MID$(S$,2):LOCATE I+
15,1,0:PRINT T$:NEXT I:RETURN
6845 '
6850 ' PLOT GRAPH LABELS ROUTINE
6900 LS=LEN(S$):C=INT((X2-X1)/8):IF LS>C THEN S$=LEFT$(S$,C)
6910 X=X1+(X2-X1+1)/2-8*LEN(S$)/2+4:Y=Y1+(Y2-Y1+1)/2-4
6920 WHILE S$<>""
6930 IF X>639 THEN 6950 ELSE C$=LEFT$(S$,1):S$=MID$(S$,2)
6940 GOSUB 6960:IF X>=0 THEN X=X+8
6950 WEND:RETURN
6960 ' PLACE CHARACTER AT PARTICULAR COORDINATES
6970 A%(1)=8:A%(2)=8:DEF SEG=&HF000:OFFSET=8*ASC(C$)+&HFA6E
6980 FOR J=1 TO 8:B$(J)=HEX$(PEEK(OFFSET+J-1))
6990 IF LEN(B$(J))=1 THEN B$(J)="0"+B$(J)
7000 NEXT J
7010 FOR J=1 TO 4:A%(J+2)=VAL("&H"+B$(2*J)+B$(2*J-1)):NEXT J
7020 DEF SEG:PUT(X,Y),A%:RETURN
7030 '
7040 ' INPUT ROUTINE for numbers, names, function keys, comments, &
Y/N answers
7110 COLOR 6,0
7120 SXFLD=XFLD:S$="":C$="":T$="":N$="":ESCK=0:GOSUB 30030:IF SND%>
1 THEN BEEP
7130 IF INFLG>2 THEN 7150
7140 C$=INKEY$:IF C$="" THEN 7140 ELSE 7170
7150 LOCATE YFLD,XFLD,1
7160 C$=INKEY$:IF C$="" THEN 7150
7170 IF C$=CHR$(27) THEN ESCK=1:RETURN ' escape key
7180 IF INFLG=2 THEN 7370
7190 IF LEN(C$)>1 THEN 7450 ' invalid
7200 C=ASC(C$)
7210 IF INFLG=1 THEN 7320
7220 IF C=8 THEN 7500
7230 IF INFLG=5 THEN 7800 ' comment
7240 IF INFLG=3 THEN 7390 ' number
7250 ' Name Routine
7260 IF C=13 THEN RETURN
7270 IF C=46 AND NUMDIG>10 THEN 7470 ' okay
7280 IF C=91 OR C=92 OR C=93 OR C=124 THEN 7450 ' invalid
7290 IF C=34 OR C=39 OR C=42 OR C=43 OR C=46 OR C=47 THEN 7450 ' in
valid
7300 IF (C>32 AND C<59) OR (C>63 AND C<127) THEN 7470 ELSE 7450

```

```

7310 'Yes/No routine
7320 IF C=89 OR C=121 THEN C$='Y':RETURN
7330 IF C=78 OR C=110 THEN C$='N':RETURN
7340 IF C=13 THEN C$='N':RETURN
7350 GOTO 7450
7360 'Function Key
7370 IF LEN(C$)<2 THEN 7450 ELSE C$=RIGHT$(C$,1):C=ASC(C$)
7380 IF (C>58 AND C<69) OR (C>83 AND C<114) THEN RETURN ELSE 7450
7390 'Number Routine
7400 IF C=13 THEN 7540 'carriage return
7410 IF C>47 AND C<58 THEN 7470 'okay
7420 IF C=46 OR C=43 OR C=45 OR C=32 THEN 7470 'okay
7430 IF C=69 OR C=101 THEN C$=CHR$(69):GOTO 7470 'okay
7440 'Illegal character
7450 GOSUB 30760:BEEP:GOTO 7130
7460 'Character okay
7470 IF LEN(S$)>=NUMDIG THEN 7450
7480 LOCATE YFLD,XFLD:PRINT C$;:S$=S$+C$:XFLD=XFLD+1:GOTO 7130
7490 'Backspace
7500 XFLD=XFLD-1:LOCATE YFLD,XFLD:PRINT" ";:IF XFLD<=SXFLD THEN XFLD=SXFLD
7510 LNT=LEN(S$)-1:IF LNT<0 THEN LNT=0
7520 S$=MID$(S$,1,LNT):GOTO 7130
7530 'Number Check routine
7540 NUMCK=0:DECPT=0:EXPNT=0:SIGN=0
7550 IF S$="" THEN 7770 ELSE T$=LEFT$(S$,1)
7560 IF T$="+" OR T$="-" THEN N$=T$:S$=MID$(S$,2)
7570 IF T$="." THEN N$=T$:DECPT=1:S$=MID$(S$,2)
7580 IF S$="" THEN 7790
7590 IF (T$="+" OR T$="-") AND INSTR(".",LEFT$(S$,1))=1 THEN DECPT=1:T$=LEFT$(S$,1):N$=N$+T$:S$=MID$(S$,2)
7600 IF INSTR(DIGIT$,LEFT$(S$,1))=0 THEN 7790
7610 WHILE S$<>"":T$=LEFT$(S$,1):S$=MID$(S$,2)
7620 IF T$="" THEN 7680
7630 IF (T$="+" OR T$="-") AND RIGHT$(N$,1)<>"E" THEN NUMCK=1:GOTO 7680
7640 IF T$="." THEN IF DECPT=1 OR EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:DECPT=1
7650 IF T$="+" OR T$="-" THEN IF SIGN=1 OR EXPNT=0 THEN NUMCK=1 ELSE N$=N$+T$:SIGN=1
7660 IF T$="E" THEN IF EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:EXPNT=1:DECPT=1
7670 IF T$<> "." AND T$<> "+" AND T$<> "-" AND T$<> "E" THEN N$=N$+T$
7680 WEND:IF NUMCK=1 THEN 7790
7690 'Check for overflow
7700 D=0:N%=INSTR(N$,"E"):IF N%=0 THEN 7770
7710 IF N%=1 THEN 7790
7720 S$=LEFT$(N$,N%-1):C$=MID$(N$,N%+1):S=VAL(S$)
7730 IF S=0 THEN D=0 ELSE D=INT(LOG(ABS(S))/LOG(10))
7740 IF C$<> "" THEN D=D+VAL(C$)
7750 IF D<-37 OR D>37 THEN 7790

```

```

7760 'convert
7770 S=VAL(N$):RETURN
7780 'invalid
7790 NUMCK=1:GOSUB 30760:BEEP:LOCATE YFLD,SXFLD,0:PRINT SPC(NUMDIG)
:XFLLD= SXFLD:GOTO 7120
7800 'Comment routine
7810 IF C$=CHR$(13) THEN RETURN
7820 IF LEN(S$)>=NUMDIG THEN RETURN ELSE 7480
7830 '
8000 'BODE PLOT ROUTINE
8030 IF PMAXX=PMINX THEN PMAXX=PMAXX+1:PMINX=PMINX-1
8032 IF PMAXY=PMINY THEN PMAXY=PMAXY+1:PMINY=PMINY-1
8090 IF PCT=0 OR (PCT=1 AND PP=0) THEN SCREEN 2:CLS:PN=PN-1
8100 PLNL=50:PLNR=589:IF PCT=0 THEN PLNT=12:PLNB=173:PNINY=5.5:GOTO
8130
8110 IF PP=1 THEN PLNT=12:PLNB=100:GOTO 8130
8120 PLNT=116:PLNB=173:PNINY=3.5
8130 PNINX=6.5:PLOCY1=INT(PLNT/8+.5):PLOCY2=PLOCY1+1
8140 PLOCY3=INT(PLNB/8+.625):PLOCY4=PLOCY3+1:PLOCY5=PLOCY4+1
8150 GOSUB 5190:IF PCT=0 OR (PCT=1 AND PP=1) THEN LOCATE 1,7 ELSE L
OCATE 14,7
8160 PRINT"SCALE X-AXIS BY 1, Y-AXIS BY";PSCLY;
8170 PLOCWY=(PLOCY3-PLOCY2+1)/PNINPY:PLNWX=(PLNB-PLNT)/PNINPY:X1=PL
NL-49:X2=PLNL-4
8180 FOR PII=0 TO PNINPY:PLNY1=PLNB-PII*PLNWX
8190 Y1=PLNY1-5:Y2=PLNY1+5:S$=STR$(PMINPY+PII*PWDY):GOSUB 6850
8200 IF PII=0 OR PII=PNINPY THEN 8220
8210 LINE(PLNL-4,PLNY1)-(PLNL,PLNY1):LINE(PLNR,PLNY1)-(PLNR+4,PLNY1
):GOTO 8230
8220 LINE(PLNL,PLNY1)-(PLNR,PLNY1)
8230 NEXT PII
8240 'X GRID AND LABELS
8250 IF INT(PMAXX)=PMAXX THEN 8280
8260 IF PMAXX>0 THEN PTMAX=INT(PMAXX) ELSE PTMAX=INT(PMAXX)-1
8270 PSMAX=10^PTMAX:PMAXX=FNLOG10((INT(10^PMAXX/PSMAX)+1)*PSMAX)
8280 IF INT(PMINX)=PMINX THEN PTMIN=PMINX:GOTO 8310
8290 IF PMINX>0 THEN PTMIN=INT(PMINX) ELSE PTMIN=INT(PMINX)-1
8300 PSMIN=10^PTMIN:PMINX=FNLOG10((INT(10^PMINX/PSMIN))*PSMIN)
8310 LINE(PLNL,PLNB)-(PLNL,PLNT):LINE(PLNR,PLNB)-(PLNR,PLNT)
8320 PPDEC=(PLNR-PLNL)/(PMAXX-PMINX):PTIC=10^PMINX
8330 IF PCT=0 OR (PCT=1 AND PP=0) THEN Y1=182 ELSE Y1=112
8340 Y2=Y1+8:PMULT=PTMIN
8350 IF PTIC>10^PMAXX THEN 8420
8360 IF PTIC>=(10^PMULT)*10 THEN PMULT=PMULT+1
8365 PTIC$=STR$(PTIC):PTIC=VAL(PTIC$)
8370 PLNX1=PLNL+(FNLOG10(PTIC)-PMINX)*PPDEC
8380 LINE(PLNX1,PLNB+3)-(PLNX1,PLNB):LINE(PLNX1,PLNT)-(PLNX1,PLNT-3
)
8390 IF (PMAXX-PMINX)>=1 AND INT(FNLOG10(PTIC))<>FNLOG10(PTIC) THEN
8410
8395 IF P=1 AND PCT=1 AND PP=1 THEN 8410

```

```

8400 IF PTIC<1 THEN S$=STR$(INT(PTIC*10*ABS(PMULT)+.4))+ "E"+STR$(PM
ULT):GOTO 8407
8405 S$=STR$(INT(PTIC/(10*ABS(PMULT))+.4))+ "E"+STR$(PMULT)
8407 X1=PLNX1-30:X2=PLNX1+25:GOSUB 6850
8410 PTIC=PTIC+10*PMULT:GOTO 8350
8420 GOSUB 6834: IF ZOOM=0 THEN GOSUB 5460 ELSE GOSUB 5500
8430 P=1: RETURN
30000 '***** SUBROUTINES *****
30020 'CLEAR KBD BUFFER
30030 DEF SEG=0: POKE 1050,PEEK(1052): DEF SEG
30040 RETURN
30050 '
30060 'OPEN DISK FILE FOR INPUT/OUTPUT
30070 'Calling routine specifies filename$, filenum, fileapp, & file
tpe
30080 'fileapp=0 no append, fileapp=1 append, filetype=0 output =1 i
nput
30090 ON ERROR GOTO 30150: IF FILEAPP=0 THEN 30110 ELSE FILEAPP=0
30100 OPEN FILENAM$ FOR APPEND AS #FILENUM:GOTO 30140
30110 IF FILETPE=1 THEN 30130
30120 OPEN FILENAM$ FOR OUTPUT AS #FILENUM:GOTO 30140
30130 OPEN FILENAM$ FOR INPUT AS #FILENUM
30140 FERR$="OK"
30145 ON ERROR GOTO 0: RETURN
30150 GOSUB 31260: RESUME 30145
30170 '
30180 'FILE DIRECTORY SUBROUTINE
30190 ON ERROR GOTO 30280
30200 COLOR 12,0:CLS:GOSUB 30030:LOCATE 1,18,0: IF SND%>1 THEN BEEP
30210 PRINT"WHICH DRIVE? (A, B, OR AB for both) >";:COLOR 6,0: INPU
T"" ,COM$
30220 PRINT:COLOR 3,0
30230 IF COM$="a:" OR COM$="A:" OR COM$="a" OR COM$="A" THEN PRINT"
A: \":FILES"A:*. *":GOTO 30310
30240 IF COM$="b:" OR COM$="B:" OR COM$="b" OR COM$="B" THEN PRINT"
B: \":FILES"B:*. *":GOTO 30310
30250 IF COM$="ab:" OR COM$="AB:" OR COM$="ab" OR COM$="AB" THEN 30
260 ELSE 30270
30260 PRINT"A: \":FILES"A:*. *":PRINT"B: \":FILES"B:*. *":GOTO 30310
30270 GOSUB 30760:GOTO 30200
30280 GOSUB 31260:GOSUB 30760:COLOR 15,0:PRINT:PRINT SPC(30) FERR$:
COLOR 7,0
30290 PRINT:PRINT SPC(31) RT$:INFLG=1:GOSUB 7120
30300 IF C$="Y" THEN RESUME 30200 ELSE RESUME 30310
30310 ON ERROR GOTO 0
30320 YFLD=24:GOSUB 31680:LOCATE 24,28,0:COLOR 12,0:PRINT"KILL ANY
FILES? (Y/N)";:INFLG=1:GOSUB 7120: IF C$="N" OR ESCK=1 THEN RETURN
30330 GOSUB 31680:LOCATE 24,32,0:PRINT"FILENAME >";:COLOR 6,0
30340 YFLD=24:XFLD=43:NUMDIG=14:INFLG=4:GOSUB 7110: IF ESCK=1 THEN R
ETURN
30350 FILENAM$=S$:GOSUB 31530: IF FERR$="OK" THEN 30320 ELSE GOSUB 3

```

```

1680:GOSUB 30760
30360 COLOR 15,0:LOCATE 24,13,0:PRINT FERR$;:LOCATE 24,47,0:COLOR 7
,0:PRINT RT$;
30370 INFLG=1:GOSUB 7120:IF C$="N" OR ESCK=1 THEN RETURN ELSE COLOR
12,0:GOTO 30330
30380 '
30390 'PROGRAM WRITE TO ARCHIVE
30400 'Calling routine provides COM$
30410 FILENAM$=ARCNAM$:FILENUM=2:FILEAPP=1:FILETPE=0
30420 GOSUB 30090:IF FERR$<>"OK" THEN CLOSE #FILENUM:GOTO 30450 'op
en file
30430 ON ERROR GOTO 30490:PRINT #FILENUM,COM$
30440 CLOSE #FILENUM:ON ERROR GOTO 0:RETURN
30450 CLS:COLOR 15,0:LOCATE 10,28,0:PRINT"ERROR ARCHIVING SESSION"
30460 LOCATE 12,28,0:PRINT FERR$:COLOR 7,0:GOSUB 30760
30470 LOCATE 14,30,0:PRINT RT$:INFLG=1:GOSUB 7120
30480 IF C$="Y" THEN 30420 ELSE RETURN
30490 ON ERROR GOTO 0:GOSUB 31260:RESUME 30450
30500 '
30510 'FILENAME INFORMATION & DIRECTORY ROUTINE
30520 COLOR 3,0:CLS
30530 LOCATE 2,15,0:PRINT"A filename is 1-8 letters/digits. A file
name may"
30540 LOCATE 3,15,0:PRINT"be preceded by a disk drive letter and a
colon,"
30550 LOCATE 4,15,0:PRINT"otherwise the current logged drive is use
d."
30560 IF FILETPE=0 THEN COLOR 15,0:LOCATE 6,8,0:PRINT"WARNING! Usi
ng an existing filename will erase the existing file."
30570 COLOR 3,0:LOCATE 8,24,0:PRINT"Would you like to see your"
30580 COLOR 12,0:LOCATE 9,26,0:PRINT"File Directory? (Y/N)"
30590 INFLG=1:GOSUB 7120:IF C$="Y" THEN GOSUB 30190
30600 RETURN
30610 '
30620 'INPUT FILENAME ROUTINE
30630 'calling routine supplies filext$, filenum, filetype, fileapp
30640 'leaves file open if OK, closes if FO=1
30650 CLS:COLOR 3,0:LOCATE 2,24,0:PRINT"Enter filename including dr
ive."
30660 LOCATE 3,20,0:PRINT"File will be given the extension "FILEXT$
30670 LOCATE 5,30,0:COLOR 12,0:PRINT"FILENAME >"
30680 XFID=41:YFID=5:NUMDIG=10:INFLG=4:GOSUB 7110:IF ESCK=1 THEN RE
TURN
30690 FILENAM$=S$+FILEXT$:GOSUB 30090
30700 IF FERR$="OK" THEN RETURN ELSE CLOSE #FILENUM
30710 COLOR 15,0:GOSUB 30760:LOCATE 7,28,0:PRINT FERR$:COLOR 7,0
30720 LOCATE 9,30,0:PRINT RT$:INFLG=1:GOSUB 7120
30730 IF C$="Y" THEN 30650 ELSE RETURN
30740 '
30750 'FLUB SOUND SUBROUTINE
30760 IF SND%<1 THEN RETURN ELSE SOUND 1000,2:SOUND 1180,3:BEEP:RET

```

```

URN
30770 '
30780 'PRESS ANY KEY ROUTINE
30790 'Caller specifies XFLD & YFLD coordinates of message
30800 COLOR 7,0:LOCATE YFLD,XFLD,0:PRINT"Press any key.....";:GOS
UB 30030
30810 IF SND%>1 THEN BEEP
30820 C$=INKEY$:IF C$="" THEN 30820
30830 IF C$=CHR$(27) THEN ESCK=1 ELSE ESCK=0
30840 RETURN
30850 COLOR 7,0:PRINT CHR$(13) CHR$(13) SPC(29)"Press any key.....
.";:GOSUB 30030:GOTO 30810
30860 '
30870 'INPUT/PRINT MATRIX ROUTINE
30880 'Caller specifies MAT, XFLD, YFLD, NUMDIG, P=1 PRINT
30890 COLOR 12,0:LOCATE YFLD,XFLD,0:PRINT COM$:YFLD=YFLD+2:XFLD=1:I
NFLG=3
30900 FOR I=1 TO ROW%:XOFF=6:NC=0:COLOR 12,0:LOCATE YFLD,XFLD:PRINT
"R";
30910 LOCATE YFLD,XFLD+1:PRINT I;:COLOR 6,0:XFLD=6:FOR J=1 TO COL%:
NC=NC+1
30920 IF P=1 THEN 30980 ELSE GOSUB 7110:IF ESCK=1 THEN RETURN
30930 IF MAT=1 THEN A(J)=S:GOTO 31040
30940 IF MAT=2 THEN B(J)=S:GOTO 31040
30980 LOCATE YFLD,XFLD,0
30990 IF MAT=1 THEN PRINT A(J);:GOTO 31040
31000 IF MAT=2 THEN PRINT B(J);:GOTO 31040
31040 XFLD=NC*(NUMDIG+2)+XOFF:IF XFLD>70 THEN YFLD=YFLD+1:XFLD=10:X
OFF=10:NC=0:IF YFLD>22 THEN GOSUB 31070
31050 NEXT J:YFLD=YFLD+1:IF YFLD>22 THEN GOSUB 31070
31060 XFLD=1:NEXT I:RETURN
31070 SXFLD=XFLD:XFLD=28:YFLD=24:GOSUB 30800:YFLD=3:XFLD=SXFLD:COLO
R 6,0:CLS:RETURN
31080 '
31090 'READ MATRIX FROM DISK FILE
31100 'Caller specifies MAT, FILENUM, ROW%, COL%, & opens file
31110 ON ERROR GOTO 31220
31120 FOR I=1 TO ROW%:FOR J=1 TO COL%:IF EOF(FILENUM) THEN 31210
31130 IF MAT=0 THEN INPUT #FILENUM,KDIM(I):GOTO 31200
31140 IF MAT=1 THEN INPUT #FILENUM,A(J):GOTO 31200
31150 IF MAT=2 THEN INPUT #FILENUM,B(J):GOTO 31200
31190 IF MAT=6 THEN INPUT #FILENUM,OP(I,J)
31200 NEXT J,I:FERR$="OK":RETURN
31210 FERR$="END OF FILE BEFORE FINISHED":RETURN
31220 ON ERROR GOTO 0:GOSUB 31260:RESUME 31230
31230 RETURN
31240 '
31250 'ERROR CHECKING ROUTINE
31260 IF ERR=64 THEN FERR$="BAD FILE NAME":RETURN
31270 IF ERR=70 THEN FERR$="DISK WRITE PROTECT":RETURN
31280 IF ERR=53 THEN FERR$="FILE NOT FOUND":RETURN

```

```

31290 IF ERR=67 THEN FERR$="TOO MANY FILES ON DISK":RETURN
31300 IF ERR=55 THEN FERR$="FILE ALREADY IN USE":RETURN
31310 IF ERR=71 THEN FERR$="DISK NOT READY":RETURN
31320 IF ERR=61 THEN FERR$="DISK FULL":RETURN
31330 IF ERR=72 THEN FERR$="BAD DISK MEDIA":RETURN
31340 FERR$="DISK I/O ERROR":RETURN
31350 '
31360 'SAVE MATRIX ON DISK
31370 'Caller specifies MAT, FILENUM, ROW%, COL% & opens file
31380 ON ERROR GOTO 31480
31390 FOR I=1 TO ROW%:FOR J=1 TO COL%
31400 IF MAT=0 THEN PRINT #FILENUM,KDIM(I);:GOTO 31470
31410 IF MAT=1 THEN PRINT #FILENUM,A(J);:GOTO 31470
31420 IF MAT=2 THEN PRINT #FILENUM,B(J);:GOTO 31470
31460 IF MAT=6 THEN PRINT #FILENUM,OP(I,J);
31470 NEXT J,I:FERR$="OK":ON ERROR GOTO 0:RETURN
31480 ON ERROR GOTO 0:GOSUB 31260:RESUME 31490
31490 RETURN
31500 '
31510 'KILL FILE ROUTINE
31520 'Caller specifies FILENAM$
31530 ON ERROR GOTO 31550:KILL FILENAM$:FERR$="OK"
31540 ON ERROR GOTO 0:RETURN
31550 GOSUB 31260:RESUME 31540
31570 '
31580 'CHANGE MATRIX ELEMENT
31590 'Caller specifies MAT, ROW1%, COL1%, & S (the new value)
31600 IF MAT=0 THEN KDIM(ROW1%)=S:RETURN
31610 IF MAT=1 THEN A(COL1%)=S:RETURN
31620 IF MAT=2 THEN B(COL1%)=S:RETURN
31660 '
31670 'ERASE ANY LINE ROUTINE
31680 LOCATE YFLD,1,0:PRINT SPC(79);:RETURN
31690 '
31700 'WRITE BLANK TO ARCHIVE
31710 COM$="":GOSUB 30410:RETURN
31720 '
32300 'NO MODEL/MATRIX ERROR ROUTINE
32310 GOSUB 30760:CLS:COLOR 15,0:LOCATE 10,28,0:PRINT COM$:XFLD=28:
YFLD=12:GOSUB 30800:RETURN
32320 '
32330 'ERASE EXISTING ERROR ROUTINE
32340 GOSUB 30760:CLS:COLOR 15,0:LOCATE 10,24,0:PRINT COM$:COLOR 7,
0:INFLG=1:GOSUB 7120:RETURN
32350 '
32430 'MODIFY ANY MATRIX ROUTINE
32440 CLS:XFLD=1:YFLD=3:NUMDIG=10:P=1:GOSUB 30890:ROW1%=1
32450 COLOR 12,0:LOCATE 25,20,0:PRINT"ELEMENT COLUMN >" SPC(12);
32470 XFLD=37:YFLD=25:NUMDIG=10:INFLG=3:GOSUB 7110:IF ESCK=1 THEN R
ETURN
32540 IF S>COL% OR S<1 THEN GOSUB 30760:GOTO 32450

```

```

32550 COL1%=S:LOCATE 25,37,0:PRINT COL1% SPC(6);:LOCATE 25,41,0:COL
OR 12,0:PRINT"VALUE >";:XFLD=49:YFLD=25
32560 GOSUB 7110:IF ESCK=1 THEN RETURN ELSE GOSUB 31600:GOTO 32440
32590 '
32700 GOSUB 30760:CLS:COLOR 15,0:LOCATE 10,28,0:PRINT COM$:COLOR 7,
0
32710 LOCATE 12,33,0:PRINT RT$:INFLG=1:GOSUB 7120:RETURN
32720 '
32770 'ROOT FINDING ROUTINE
32780 'N=order of polynomial,SDI=matrix of coefficients
32790 IF N<5 THEN 34000
32795 IREV=1:NC=N+1:FOR I=1 TO NC:SDJ(I)=SDI(I):NEXT I:P=0:Q=0:R=0
32800 IF SDJ(1)<>0 THEN 32820 ELSE NC=NC-1
32810 SP0(NC,1)=0:SP0(NC,2)=0:FOR I=1 TO NC:SDJ(I)=SDJ(I+1):NEXT I:
GOTO 32800
32820 IF (NC-1)=0 THEN RETURN
32830 IF (NC-2)=0 THEN R=-SDJ(1)/SDJ(2):GOTO 33050
32840 IF (NC-3)=0 THEN P=SDJ(2)/SDJ(3):Q=SDJ(1)/SDJ(3):GOTO 33070
32850 IF (ABS(SDJ(NC-1)/SDJ(NC))-ABS(SDJ(2)/SDJ(1)))>=0 THEN 32900
32860 IREV=-IREV:FOR I=1 TO NC/2:NL=NC+1-I:TEMP=SDJ(NL):SDJ(NL)=SDJ
(I):SDJ(I)=TEMP:NEXT I
32870 IF Q=0 THEN P=0:GOTO 32890
32880 P=P/Q:Q=1/Q
32890 IF R<>0 THEN R=1/R
32900 E=5E-10:SG(NC)=SDJ(NC):SDI(NC)=SDJ(NC):SG(NC+1)=0:SDI(NC+1)=0
:NP=NC-1
32910 FOR J=1 TO 1000:FOR II=1 TO NP:I=NC-II:SG(I)=SDJ(I)+R*SG(I+1)
32920 SDI(I)=SG(I)+R*SDI(I+1):NEXT II
32930 IF (ABS(SG(1)/SDJ(1))-E)<=0 THEN 33050
32940 IF SDI(2)=0 THEN R=R+1 ELSE R=R-SG(1)/SDI(2)
32950 FOR II=1 TO NP:I=NC-II
32960 SG(I)=SDJ(I)-P*SG(I+1)-Q*SG(I+2):SDI(I)=SG(I)-P*SDI(I+1)-Q*SD
I(I+2):NEXT II
32970 IF SDJ(2)<>0 THEN 32990
32980 IF ABS(SG(2)/SDJ(1))-E<=0 THEN 33000 ELSE 33010
32990 IF ABS(SG(2)/SDJ(2))-E>0 THEN 33010
33000 IF ABS(SG(1)/SDJ(1))-E<=0 THEN 33070
33010 CBAR=SDI(2)-SG(2):D=SDI(3)*2-CBAR*SDI(4)
33020 IF D=0 THEN P=P-2:Q=Q*(Q+1):GOTO 33040
33030 P=P+(SG(2)*SDI(3)-SG(1)*SDI(4))/D:Q=Q+(-SG(2)*CBAR+SG(1)*SDI(
3))/D
33040 NEXT J:E=E*10:GOTO 32910
33050 NC=NC-1:SP0(NC,2)=0:IF IREV<0 THEN SP0(NC,1)=1/R ELSE SP0(NC,
1)=R
33060 FOR I=1 TO NC:SDJ(I)=SG(I+1):NEXT I:GOTO 32820
33070 NC=NC-2:IF IREV<0 THEN QP=1/Q:PP=P/(Q*2):GOTO 33090
33080 QP=Q:PP=P/2
33090 TEMP=PP*2-QP:IF TEMP>=0 THEN 33110
33100 SP0(NC+1,1)=-PP:SP0(NC,1)=-PP:SP0(NC+1,2)=SQR(-TEMP):SP0(NC,2
)=-SP0(NC+1,2):GOTO 33130
33110 IF PP=0 THEN SP0(NC+1,1)=-SQR(TEMP) ELSE SP0(NC+1,1)=(-PP/ABS

```

```

(PP))*(ABS(PP)+SQR(TEMP))
33120 SPO(NC+1,2)=0: SPO(NC,1)=QP/SPO(NC+1,1): SPO(NC,2)=0
33130 FOR I=1 TO NC: SDJ(I)=SG(I+2): NEXT I: GOTO 32820
33140 '
33150 'NICHOLS TABLE SEARCH
33160 NE=.005: NS=0
33170 NP=0: FOR I=1 TO KDIM(3): IF OP(I,2)=0 THEN 33190
33175 IF ABS((PYCOD-OP(I,2))/OP(I,2))>NE THEN 33190
33180 NP=NP+1: IF NP>10 THEN 33200 ELSE SNG(NP)=I
33190 NEXT I
33200 IF NS<>0 THEN 33240 ELSE NS=1
33210 IF NP=0 THEN NPATH=1: GOTO 33250
33220 IF NP>10 THEN NPATH=2: GOTO 33260
33230 GOTO 33275
33240 IF NPATH=2 THEN 33260
33250 IF NP<>0 THEN 33270 ELSE NE=NE*10: IF NE<=.05 THEN 33170 ELSE
FF=0: RETURN
33260 IF NP>10 THEN NE=NE/10: IF NE>.0005 THEN 33170 ELSE 33330
33270 IF NP>10 THEN 33330
33275 NE=.05: FOR I=1 TO NP: IF OP(SNG(I),3)=0 THEN 33290
33277 NTE=ABS((PXCOD-OP(SNG(I),3))/OP(SNG(I),3))
33280 IF NTE<NE THEN NE=NTE: NPT=I
33290 NEXT I: IF NE=.05 THEN FF=0: RETURN
33300 W=OP(SNG(NPT),1): FF=1
33305 GOSUB 6380: PYCOD=OP(SNG(NPT),2): PXCOD=OP(SNG(NPT),3)
33310 CRSRX=FNPSCLX(PXCOD/PSCLX,PMAXPX,PMINPX,PLNR,PLNL)
33320 CRSRY=FNPSCLY(PYCOD/PSCLY,PMAXPY,PMINPY,PLNT,PLNB): GOSUB 6300: R
ETURN
33330 NPP=0: FOR I=1 TO KDIM(3): IF OP(I,3)=0 THEN 33350
33335 IF ABS((PXCOD-OP(I,3))/OP(I,3))>NE THEN 33350
33340 NPP=NPP+1: IF NPP>10 THEN 33360
33350 NEXT I: IF NPP<10 THEN 33275
33360 FOR I=1 TO 10: NPP=SNG(I)-(SNG(1)-1): NEXT I: IF NPP<>55 THEN 33
275
33380 FOR I=SNG(1) TO KDIM(3): IF OP(I,2)=0 THEN 33400
33390 IF ABS((PYCOD-OP(I,2))/OP(I,2))>NE THEN SNG(2)=I-1: GOTO 33410
33400 NEXT I: SNG(2)=KDIM(3)
33410 W1=OP(SNG(1),1): W2=OP(SNG(2),1): NPT=SNG(1): FF=2: GOTO 33305
33999 '
34000 IF N=1 THEN SPO(N,1)=-SDI(N): SPO(N,2)=0: RETURN
34010 IF N=3 THEN 34070
34020 IF N=4 THEN 34200
34030 SR=SDI(2)*2-4*SDI(1): SI=0: N=2: GOSUB 34610
34040 IF RI<>0 THEN 34060 ELSE SPO(1,2)=0: SPO(2,2)=0
34050 SPO(1,1)=(-SDI(2)+RR)/2: SPO(2,1)=(-SDI(2)-RR)/2: RETURN
34060 SPO(1,1)=-SDI(2)/2: SPO(2,1)=SPO(1,1): SPO(1,2)=RI/2: SPO(2,2)=-
RI/2: RETURN
34070 A=(3*SDI(2)-SDI(3)*2)/3: B=(2*SDI(3)*3-9*SDI(3)*SDI(2)+27*SDI(
1))/27
34080 SR=B*2/4+A*3/27: SI=0: N=2: GOSUB 34610
34090 QTR=RR: QTI=RI: SR=QTR-B/2: SI=QTI: N=3

```

```

34100 IF SR<0 AND SI=0 THEN SR=-SR:GOSUB 34610:RR=-RR:GOTO 34120
34110 GOSUB 34610
34120 QAR=RR:QAI=RI:SR=-QTR-B/2:SI=-QTI
34130 IF SR<0 AND SI=0 THEN SR=-SR:GOSUB 34610:RR=-RR:GOTO 34150
34140 GOSUB 34610
34150 QBR=RR:QBI=RI:SR=-3:SI=0:N=2:GOSUB 34610
34160 SP0(1,1)=QAR+QBR:SP0(1,2)=QAI+QBI:QTR=(QAR-QBR)/2:QTI=(QAI-QB
I)/2
34170 SR=FNCRM(QTR,QTI,RR,RI):SI=FNCIM(QTR,QTI,RR,RI)
34180 SSR=-(QAR+QBR)/2:SSI=-(QAI+QBI)/2:SP0(2,1)=SSR+SR:SP0(2,2)=SS
I+SI:SP0(3,1)=SSR-SR:SP0(3,2)=SSI-SI
34190 FOR I=1 TO 3:SP0(I,1)=SP0(I,1)-SDI(3)/3:NEXT I:RETURN
34200 A=SDI(4):B=SDI(3):C=SDI(2):D=SDI(1)
34210 FOR I=1 TO 4:SP0(I,1)=0:SP0(I,2)=0:NEXT I:PI=3.1415926#:PIO2=
1.5707963#
34220 QP=-B:QQ=A*C-4*D:QR=-(A^2)*D+4*B*D-C^2
34230 QCA=(3*QQ-QP^2)/3:QCB=(2*QP^3-9*QP*QQ+27*QR)/27
34240 QTEMP=QCB^2/4+QCA^3/27:N=2:SR=QTEMP:SI=0
34250 GOSUB 34610:QTR=RR:QTI=RI
34260 SR=QTR-QCB/2:SI=QTI:N=3
34270 IF SR<0 AND SI=0 THEN SR=-SR:GOSUB 34610:RR=-RR:GOTO 34290
34280 GOSUB 34610
34290 QCAAR=RR:QCAAI=RI:SR=-QTR-QCB/2:SI=-QTI
34300 IF SR<0 AND SI=0 THEN SR=-SR:GOSUB 34610:RR=-RR:GOTO 34320
34310 GOSUB 34610
34320 YR=QCAAR+RR-QP/3:YI=QCAAI+RI
34330 SR=A^2/4-B+YR:SI=YI:N=2:GOSUB 34610
34340 QRR=RR:QRI=RI
34350 IF QRR=0 AND QRI=0 THEN 34450
34360 QTEMP=4*A*B-8*C-A^3
34370 QTR=FNCRD(QTEMP,0,4*QRR,4*QRI)
34380 QTI=FNCID(QTEMP,0,4*QRR,4*QRI)
34390 QRTR=FNCRM(QRR,QRI,QRR,QRI)
34400 QRTI=FNCIM(QRR,QRI,QRR,QRI)
34410 SR=3*A^2/4-QRTR-2*B+QTR:N=2:SI=-QRTI+QTI:GOSUB 34610
34420 QDR=RR:QDI=RI
34430 SR=3*A^2/4-QRTR-2*B-QTR:N=2:SI=-QRTI-QTI:GOSUB 34610
34440 QER=RR:QEI=RI:GOTO 34500
34450 QTR=FNCRM(YR,YI,YR,YI)-4*D
34460 QTI=FNCIM(YR,YI,YR,YI)
34470 SR=QTR:SI=QTI:N=2:GOSUB 34610:QTR=2*RR:QTI=2*RI
34480 SR=QTR+3*A^2/4-2*B:SI=QTI:GOSUB 34610:QDR=RR:QDI=RI
34490 SR=-QTR+3*A^2/4-2*B:SI=-QTI:GOSUB 34610:QER=RR:QEI=RI
34500 SP0(1,1)=-A/4+QRR/2+QDR/2
34510 SP0(1,2)=QRI/2+QDI/2
34520 SP0(2,1)=-A/4+QRR/2-QDR/2
34530 SP0(2,2)=QRI/2-QDI/2
34540 SP0(3,1)=-A/4-QRR/2+QER/2
34550 SP0(3,2)=-QRI/2+QEI/2
34560 SP0(4,1)=-A/4-QRR/2-QER/2
34570 SP0(4,2)=-QRI/2-QEI/2

```

```

34580 RETURN
34590 '
34600 'This subroutine finds the Nth root of a complex number.
34610 PI=3.1415926#:PIO2=1.5707963#
34620 IF SI=0 AND SR=0 THEN RI=0:RR=0:ANGLE=0:RETURN
34630 IF SI<0 AND SR<0 THEN ANGLE=-PI+ATN(SI/SR):GOTO 34710
34640 IF SI<0 AND SR=0 THEN ANGLE=-PIO2:GOTO 34710
34650 IF SI<0 AND SR>0 THEN ANGLE=-ATN(-SI/SR):GOTO 34710
34660 IF SI=0 AND SR<0 THEN ANGLE=PI:GOTO 34710
34670 IF SI=0 AND SR>0 THEN ANGLE=0:GOTO 34710
34680 IF SI>0 AND SR<0 THEN ANGLE=PI-ATN(-SI/SR):GOTO 34710
34690 IF SI>0 AND SR=0 THEN ANGLE=PIO2:GOTO 34710
34700 ANGLE=ATN(SI/SR)
34710 NR=SQR(SI2+SR2):RR=NR(1/N)*COS(ANGLE/N):RI=NR(1/N)*SIN(AN
GLE/N)
34720 RETURN

```

C.4 SIM 3/6/85

```

10 COMMON HELP%,HELP$,ARC%,SND%,QW,ARCNAM$: OPTION BASE 1: DEFINT I,J
30 SIM$="-----SYSTEM SIMULATION-----"
40 DIM A(20,20),B(20,20),C(21,20),D(21,20),SX(20),X0(20),SP1(20,20)
,SG(20,20)
50 DIM SW(20,20),SP0(21,20),OP(1001,6),PX(1001),PY(1001),SU(20,6),S
DI(20,20)
60 DIM KDIM(6),A%(10),B$(10)
65 IF QW<1 THEN HELP%=0: HELP$='OFF': ARC%=0: SND%=1
70 MAXDIM=20: MAXPTS=1000: MAXOPT=5: DIGIT$='1234567890': RT$='RETRY?'
(Y/N)"
80 EM=0: IF ARC%=1 THEN GOSUB 31710: COM$=SIM$: GOSUB 30410: GOSUB 3171
0
90 CLS: WIDTH 80: COLOR 14,0: LOCATE 3,29,0: PRINT "SYSTEM SIMULATION"
100 COLOR 10,0: LOCATE 6,23,0: PRINT "<F1>.....ENTER NEW MODEL"
110 LOCATE 7,23,0: PRINT "<F2>.....RECALL OLD MODEL"
120 LOCATE 8,23,0: PRINT "<F3>.....MODIFY CURRENT MODEL"
130 LOCATE 9,23,0: PRINT "<F4>.....STORE MODEL ON DISK"
140 LOCATE 10,23,0: PRINT "<F5>.....ENTER INPUT PARAMETERS"
150 LOCATE 11,23,0: PRINT "<F6>.....CALCULATE DATA"
160 LOCATE 12,23,0: PRINT "<F7>.....GRAPH DATA"
170 LOCATE 13,23,0: PRINT "<F8>.....LIST/SAVE DATA"
180 LOCATE 14,23,0: PRINT "<F9>.....FILE DIRECTORY"
190 LOCATE 15,23,0: PRINT "<ESC>.....EXIT TO MAIN MENU"
200 LOCATE 16,23,0: PRINT "<Alt-F1>....HELP LEVEL"
210 LOCATE 17,23,0: PRINT "<Alt-F2>....SOUND LEVEL"
220 COLOR 14,0: LOCATE 16,46,0: PRINT HELP$: LOCATE 17,46,0: PRINT SND%
230 COLOR 10,0: LOCATE 18,23,0: PRINT "<Alt-F3>....ARCHIVE/COMMENTS"
240 COLOR 14,0: LOCATE 21,24,0: PRINT "PRESS DESIRED FUNCTION KEY"
250 INFLG=2: GOSUB 7120: IF ESCK=1 AND QW=1 THEN CHAIN "DACS"
255 IF ESCK=1 AND QW<>1 THEN 383      'system
260 IF C=59 THEN GOSUB 850: GOTO 90    'new model
270 IF C=60 THEN GOSUB 1540: GOTO 90   'recall model
280 IF C=61 THEN GOSUB 4260: GOTO 90   'modify
290 IF C=62 THEN GOSUB 1300: GOTO 90   'store model
300 IF C=63 THEN GOSUB 1900: GOTO 90   'input params
310 IF C=64 THEN GOSUB 2500: GOTO 90   'run
320 IF C=65 THEN GOSUB 2820: GOTO 90   'graph
330 IF C=66 THEN GOSUB 3590: GOTO 90   'list
340 IF C=67 THEN GOSUB 30190: GOTO 90  'files
350 IF C=104 THEN GOSUB 590: GOTO 220  'help
360 IF C=105 THEN GOSUB 550: GOTO 220  'help
370 IF C=106 THEN GOSUB 390: GOTO 90   'archive
380 GOSUB 30760: GOTO 250
381 '
382 'SYSTEM
383 CLEAR: WIDTH 80: SCREEN 0,0: COLOR 7,0: CLOSE: END
385 '
390 'ARCHIVE
392 IF ARC%=1 THEN GOSUB 420: RETURN

```

```

394 GOSUB 420: IF ARC%=0 THEN RETURN ELSE GOSUB 31710: COM$=SIM$: GOSUB
B 30410: GOSUB 31710: RETURN
396 '
410 ' ARCHIVE SESSION / ADD COMMENTS
420 COLOR 12,0: CLS: IF ARC%=1 THEN 470 ELSE LOCATE 2,27,0
430 PRINT "ARCHIVE SESSION? (Y/N)": INFLG=1: GOSUB 7120: IF ESCK=1 OR
C$="N" THEN RETURN
440 FILETPE=0: IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
450 FILEXT$=".ARC": FILENUM=2: GOSUB 30650: CLOSE #FILENUM
460 IF ESCK=1 OR FERR$<>"OK" THEN RETURN ELSE ARCNAM$=FILENAM$: ARC%
=1: CLS
470 COLOR 12,0: LOCATE 2,28,0: PRINT "ENTER COMMENTS? (Y/N)": INFLG=1:
GOSUB 7120
480 IF C$="N" OR ESCK=1 THEN RETURN ELSE NUMDIG=78: INFLG=5: XF LD=1
490 CLS: LOCATE 2,19,0: YFLD=4
500 PRINT "ENTER COMMENTS BELOW....PRESS ESC TO QUIT": PRINT
510 GOSUB 7110: IF ESCK=1 THEN GOSUB 31710: RETURN
520 COM$=S$: GOSUB 30410: IF FERR$<>"OK" THEN RETURN ELSE YFLD=YFLD+1
: XF LD=1: IF YFLD>24 THEN 490 ELSE 510
530 '
540 ' SOUND LEVEL
550 SND%=SND%+1: IF SND%>2 THEN SND%=0
560 RETURN
570 '
580 ' SET HELP LEVEL
590 IF HELP%=1 THEN HELP%=0: HELP$="OFF": RETURN
600 HELP%=1: HELP$="ON ": RETURN
610 '
840 ' ENTER NEW MODEL ROUTINE
850 IF EM=1 THEN COM$="ERASE EXISTING MODEL? (Y/N)": GOSUB 32340: IF
C$="N" OR ESCK=1 THEN RETURN
880 CLS: IF HELP%=0 THEN 990
890 COLOR 3,0: LOCATE 2,12,0: PRINT "You will be prompted for the syst
em dimension (DA), the"
900 LOCATE 3,12,0: PRINT "number of outputs (DY), the number of input
s (DU) and the"
910 LOCATE 4,12,0: PRINT "constant matrices A, B, C, D, and X0, where
"
920 LOCATE 6,31,0: COLOR 15,0: PRINT "x dot = Ax + Bu"
930 LOCATE 7,31,0: PRINT "    y = Cx + Du": COLOR 3,0
940 LOCATE 9,9,0: PRINT "X0 is the initial condition vector and the m
atrices have dimensions"
950 COLOR 15,0: LOCATE 11,34,0: PRINT "A(DA,DA)
960 LOCATE 12,34,0: PRINT "B(DA,DU)": LOCATE 13,34,0: PRINT "C(DY,DA)"
970 LOCATE 14,34,0: PRINT "D(DY,DU)": LOCATE 15,34,0: PRINT "X0(DA)"
980 XF LD=30: YFLD=17: GOSUB 30800: IF ESCK=1 THEN GOSUB 1250: RETURN
990 CLS: GOSUB 1030: IF ESCK=1 THEN GOSUB 1250: RETURN
1000 IF C$="Y" THEN 990 ELSE IF ESCK=1 THEN GOSUB 1250: RETURN
1010 GOSUB 1160: IF ESCK=1 THEN GOSUB 1250: RETURN
1020 EM=1: IF ARC%=0 THEN RETURN ELSE GOSUB 31710: COM$="-ENTER NEW M
ODEL-": GOSUB 30410: RETURN

```

```

1030 COLOR 12,0: LOCATE 6,29,0: PRINT"SYSTEM DIMENSION >"SPC(38)
1040 XFLD=49: YFLD=6: NUMDIG=12: INFLG=3: GOSUB 7110: IF ESCK=1 THEN RET
URN
1050 S=INT(S): IF S<=MAXDIM AND S>0 THEN KDIM(1)=S: GOTO 1070
1060 GOSUB 30760: COLOR 15,0: LOCATE 8,18,0: PRINT"ERROR! Value must
lie in the range 1 to"MAXDIM: GOTO 1030
1070 COLOR 12,0: LOCATE 10,29,0: PRINT"NUMBER OF INPUTS >"SPC(38): XFL
D=49
1080 YFLD=10: GOSUB 7110: IF ESCK=1 THEN RETURN
1090 S=INT(S): IF S<=MAXDIM AND S>=0 THEN KDIM(2)=S: GOTO 1110
1100 GOSUB 30760: COLOR 15,0: LOCATE 12,18,0: PRINT"ERROR! Value must
lie in the range 0 to"MAXDIM: GOTO 1070
1110 COLOR 12,0: LOCATE 14,29,0: PRINT"NUMBER OUTPUTS >"SPC(37): XFL
D=49
1120 YFLD=14: GOSUB 7110: IF ESCK=1 THEN RETURN
1130 S=INT(S): IF S<=MAXOPT AND S>0 THEN KDIM(3)=S: COLOR 7,0: GOTO 11
50
1140 GOSUB 30760: COLOR 15,0: LOCATE 16,18,0: PRINT"ERROR! Value must
lie in the range 1 to"MAXOPT: GOTO 1110
1150 LOCATE 18,23,0: PRINT"RE-ENTER ANY OF THE ABOVE? (Y/N)": INFLG=
1: GOSUB 7120: RETURN
1160 CLS: NUMDIG=10: INFLG=3: P=0: IF HELP%=0 THEN 1180
1170 LOCATE 1,6,0: PRINT"ENTER MATRICES BELOW BY ROWS PRESSING <ENTE
R> AFTER EACH ELEMENT."
1180 MAT=1: COM$="A MATRIX": XFLD=1: YFLD=3: ROW%=KDIM(1): COL%=KDIM(1):
GOSUB 30890: IF ESCK=1 THEN RETURN
1190 IF KDIM(2)=0 THEN 1210
1200 CLS: MAT=2: COM$="B MATRIX": XFLD=1: YFLD=3: ROW%=KDIM(1): COL%=KDIM
(2): GOSUB 30890: IF ESCK=1 THEN RETURN
1210 CLS: MAT=3: COM$="C MATRIX": XFLD=1: YFLD=3: ROW%=KDIM(3): COL%=KDIM
(1): GOSUB 30890: IF ESCK=1 THEN RETURN
1220 IF KDIM(2)=0 THEN 1240
1230 CLS: MAT=4: COM$="D MATRIX": XFLD=1: YFLD=3: ROW%=KDIM(3): COL%=KDIM
(2): GOSUB 30890: IF ESCK=1 THEN RETURN
1240 CLS: MAT=5: COM$="XO MATRIX": XFLD=1: YFLD=3: ROW%=KDIM(1): COL%=1: G
OSUB 30890: RETURN
1250 COM$="MODEL NOT ENTERED!!": GOSUB 32310: EM=0: RETURN
1270 '
1280 'SAVE MODEL ROUTINE
1290 'Saves current model on disk
1300 IF EM=1 THEN 1330 ELSE COM$="NO MODEL TO SAVE!!": GOSUB 32310: R
ETURN
1330 FILETPE=0: IF HELP%=1 THEN GOSUB 30520 ELSE CLS: GOSUB 30580
1340 FILEXT$=".SMD": FILENUM=1: GOSUB 30650
1350 IF ESCK=0 AND FERR$="OK" THEN 1380
1360 COM$="MODEL NOT SAVED!!": GOSUB 32310: RETURN
1380 SFILE$=FILENAM$: CLS: COLOR 3,0: LOCATE 10,30,0: PRINT"SAVING MODE
L....."
1390 IF ARC%=1 THEN GOSUB 31710: COM$="-STORE MODEL ON DISK-": GOSUB
30410: FILENUM=1
1400 MAT=0: ROW%=4: COL%=1: GOSUB 31380: IF FERR$<>"OK" THEN 1480

```

```

1410 MAT=1: ROW%=KDIM(1): COL%=KDIM(1): GOSUB 31380: IF FERR$<>"OK" THE
N 1480
1420 MAT=2: ROW%=KDIM(1): COL%=KDIM(2): GOSUB 31380: IF FERR$<>"OK" THE
N 1480
1430 MAT=3: ROW%=KDIM(3): COL%=KDIM(1): GOSUB 31380: IF FERR$<>"OK" THE
N 1480
1440 MAT=4: ROW%=KDIM(3): COL%=KDIM(2): GOSUB 31380: IF FERR$<>"OK" THE
N 1480
1450 MAT=5: ROW%=KDIM(1): COL%=1: GOSUB 31380: IF FERR$<>"OK" THEN 1480
1460 CLOSE #FILENAME: IF ARC%=1 THEN COM$="Model saved under filename
"+SFILE$: GOSUB 30410
1470 RETURN
1480 CLS: LOCATE 10,28,0: COLOR 15,0: PRINT FERR$: COLOR 7,0: GOSUB 3076
0
1490 FILENAM$=SFILE$: GOSUB 31530: LOCATE 12,33,0: PRINT RT$: INFLG=1: G
OSUB 7120
1500 IF C$="Y" AND ESCK=0 THEN 1330
1510 IF ARC%=0 THEN 1360 ELSE COM$=FERR$: GOSUB 30410: GOTO 1360
1520 '
1530 'RECALL MODEL ROUTINE
1540 IF EM=1 THEN COM$="ERASE EXISTING MODEL? (Y/N)": GOSUB 32340: I
F C$="N" OR ESCK=1 THEN RETURN
1570 FILETPE=1: IF HELP%=1 THEN GOSUB 30520 ELSE CLS: GOSUB 30580
1580 FILEXT$=".SMD": FILENUM=1: GOSUB 30650
1590 IF ESCK=0 AND FERR$="OK" THEN SFILE$=FILENAM$: GOTO 1620
1600 COM$="MODEL NOT RECALLED!!": GOSUB 32310: RETURN
1620 CLS: COLOR 3,0: LOCATE 10,30,0: PRINT"RECALLING MODEL....."
1630 IF ARC%=1 THEN GOSUB 31710: COM$="-RECALL MODEL-": GOSUB 30410: F
ILENUM=1
1640 ROW%=4: COL%=1: MAT=0: GOSUB 31110: IF FERR$<>"OK" THEN 1840
1650 ROW%=KDIM(1): COL%=KDIM(1): MAT=1: GOSUB 31110: IF FERR$<>"OK" THE
N 1840
1660 ROW%=KDIM(1): COL%=KDIM(2): MAT=2: GOSUB 31110: IF FERR$<>"OK" THE
N 1840
1670 ROW%=KDIM(3): COL%=KDIM(1): MAT=3: GOSUB 31110: IF FERR$<>"OK" THE
N 1840
1680 ROW%=KDIM(3): COL%=KDIM(2): MAT=4: GOSUB 31110: IF FERR$<>"OK" THE
N 1840
1690 ROW%=KDIM(1): COL%=1: MAT=5: GOSUB 31110: IF FERR$<>"OK" THEN 1840
ELSE EM=1
1700 CLOSE #FILENAME: COLOR 7,0: LOCATE 14,30,0: PRINT"SEE MODEL? (Y/N
)"
1710 INFLG=1: GOSUB 7120: IF C$="Y" AND ESCK=0 THEN 1740
1720 IF ARC%=1 THEN COM$="Model recalled from file "+SFILE$: GOSUB 3
0410
1730 RETURN
1740 CLS: COLOR 12,0: LOCATE 5,1,0: PRINT"SYSTEM DIMENSION >"SPC(8)"NU
MBER OF INPUTS >"SPC(8)"NUMBER OUTPUTS >"
1750 COLOR 6,0: LOCATE 5,20,0: PRINT KDIM(1): LOCATE 5,46,0: PRINT KDIM
(2): LOCATE 5,72,0: PRINT KDIM(3)
1760 GOSUB 30850: IF ESCK=1 THEN 1720 ELSE P=1: NUMDIG=10: CLS

```

```

1770 YFLD=3: XFLD=1: MAT=1: COM$='A MATRIX': ROW%=KDIM(1): COL%=KDIM(1):
GOSUB 30890: GOSUB 30850: IF ESCK=1 THEN 1720
1780 IF KDIM(2)=0 THEN 1800 ELSE CLS
1790 YFLD=3: XFLD=1: MAT=2: COM$='B MATRIX': ROW%=KDIM(1): COL%=KDIM(2):
GOSUB 30890: GOSUB 30850: IF ESCK=1 THEN 1720
1800 CLS: YFLD=3: XFLD=1: MAT=3: COM$='C MATRIX': ROW%=KDIM(3): COL%=KDIM
(1): GOSUB 30890: GOSUB 30850: IF ESCK=1 THEN 1720
1810 IF KDIM(2)=0 THEN 1830 ELSE CLS
1820 YFLD=3: XFLD=1: MAT=4: COM$='D MATRIX': ROW%=KDIM(3): COL%=KDIM(2):
GOSUB 30890: GOSUB 30850: IF ESCK=1 THEN 1720
1830 CLS: YFLD=3: XFLD=1: MAT=5: COM$='XO MATRIX': ROW%=KDIM(1): COL%=1: G
OSUB 30890: GOSUB 30850: GOTO 1720
1840 CLOSE #FILENAME: CLS: COLOR 15,0: GOSUB 30760: LOCATE 10,28,0: PRINT
FERR$
1850 LOCATE 12,33,0: COLOR 7,0: PRINT RT$: INFLG=1: GOSUB 7120
1860 IF C$='Y' AND ESCK=0 THEN 1570 ELSE EM=0
1870 IF ARC%=1 THEN COM$=FERR$+" for file "+SFILE$: GOSUB 30410
1880 GOTO 1600
1890 '
1900 ' INPUT PARAMETERS
1910 IF EM=1 THEN 1940 ELSE COM$='ENTER MODEL FIRST!!': GOSUB 32310:
RETURN
1940 CLS: IP=0: NUMDIG=12: IF HELP%=0 THEN 2010 ELSE COLOR 3,0
1950 LOCATE 2,12,0: PRINT "You will be prompted for maximum time, tim
e increment,"
1960 LOCATE 3,12,0: PRINT "and type of inputs. Maximum time divided
by time"
1970 LOCATE 4,12,0: PRINT "increment equals number of data points to
be calculated."
1980 LOCATE 5,12,0: PRINT "If TMAX/TINT > 1000, all data points will
be calculated,"
1982 LOCATE 6,12,0: PRINT "but only every ith one will be saved to st
ay below the"
1984 LOCATE 7,12,0: PRINT "1000 memory limit.": GOSUB 30850: IF ESCK=1
THEN RETURN ELSE CLS
2010 COLOR 12,0: LOCATE 1,30,0: PRINT "MAXIMUM TIME >" SPC(30): INFLG=3
2020 YFLD=1: XFLD=46: NUMDIG=12: GOSUB 7110: IF ESCK=1 THEN 2460 ELSE K
DIM(5)=S
2030 IF KDIM(5)<=0 THEN GOSUB 30760: GOTO 2010
2040 COLOR 12,0: LOCATE 3,30,0: PRINT "TIME INTERVAL >" SPC(30)
2050 YFLD=3: XFLD=46: GOSUB 7110: IF ESCK=1 THEN 2460 ELSE T=S
2060 IF T<=0 THEN GOSUB 30760: GOTO 2040
2070 S$=STR$(KDIM(5)): GOSUB 7700: ETMAX=D: S$=STR$(T): GOSUB 7700: ET=D
2080 ETT=ETMAX-ET: IF ETT<-37 OR ETT>37 THEN GOSUB 30760: YFLD=3: GOSU
B 31680: GOTO 2010
2120 IF KDIM(2)=0 THEN 2220
2130 COLOR 3,0: LOCATE 5,28,0: PRINT "<Alt-F4>...NULL INPUT" SPC(10)
2140 LOCATE 6,28,0: PRINT "<Alt-F5>...STEP INPUT"
2150 LOCATE 7,28,0: PRINT "<Alt-F6>...RAMP INPUT"
2160 LOCATE 8,28,0: PRINT "<Alt-F7>...SINE INPUT"
2170 LOCATE 9,19,0: PRINT "INPUTS NOT ASSIGNED WILL HAVE NULL VALUE"

```

```

2180 LOCATE 11,22,0:PRINT"HIT <ENTER> ON INPUT NUMBER TO END"
2190 FOR I=1 TO KDIM(2):SU(I,1)=0:NEXT I
2200 COLOR 12,0:LOCATE 13,30,0:PRINT"INPUT NUMBER    >"SPC(30):XFLD
=48:YFLD=13
2210 INFLG=3:GOSUB 7110:IF ESCK=1 THEN 2460 ELSE I=S:IF I<>0 THEN 2
240
2220 IF ARC%=1 THEN GOSUB 31710:COM$="-INPUT PARAMETERS-":GOSUB 304
10
2230 IP=1:RETURN
2240 IF I<0 OR I>KDIM(2) THEN GOSUB 30760:GOTO 2200
2250 COLOR 12,0:LOCATE 15,30,0:PRINT"PICK INPUT TYPE >":INFLG=2
2260 GOSUB 7120:IF ESCK=1 THEN 2460 ELSE COLOR 6,0
2270 IF C=107 THEN LOCATE 15,48,0:PRINT"NULL":INFLG=3:GOTO 2450
2280 IF C=108 THEN LOCATE 15,48,0:PRINT"STEP":SU(I,1)=1:GOTO 2320
2290 IF C=109 THEN LOCATE 15,48,0:PRINT"RAMP":SU(I,1)=2:GOTO 2320
2300 IF C=110 THEN LOCATE 15,48,0:PRINT"SINE":SU(I,1)=3:GOTO 2320
2310 GOSUB 30760:GOTO 2260
2320 COLOR 12,0:LOCATE 17,30,0:IF SU(I,1)=2 THEN PRINT"ZERO OFFSET
>" ELSE PRINT"INPUT AMPLITUDE >"
2330 INFLG=3:XFLD=48:YFLD=17:NUMDIG=12:GOSUB 7110:IF ESCK=1 THEN 24
60
2340 SU(I,2)=S:IF SU(I,1)=1 THEN 2370
2350 COLOR 12,0:LOCATE 19,30,0:IF SU(I,1)=2 THEN PRINT"RAMP SLOPE
>" ELSE PRINT"FREQ (rad/sec) >"
2360 XFLD=48:YFLD=19:GOSUB 7110:IF ESCK=1 THEN 2460 ELSE SU(I,3)=S
2370 COLOR 12,0:LOCATE 21,30,0:PRINT"STARTING TIME    >"SPC(30)
2380 XFLD=48:YFLD=21:GOSUB 7110:IF ESCK=1 THEN 2460
2390 IF S<0 OR S>KDIM(5) THEN GOSUB 30760:GOTO 2370
2400 SU(I,4)=S
2410 COLOR 12,0:LOCATE 23,30,0:PRINT"ENDING TIME      >"SPC(30)
2420 XFLD=48:YFLD=23:GOSUB 7110:IF ESCK=1 THEN 2460
2430 IF S<SU(I,4) THEN GOSUB 30760:GOTO 2410
2440 IF S>KDIM(5) THEN SU(I,5)=KDIM(5) ELSE SU(I,5)=S
2450 FOR YFLD=25 TO 15 STEP -2:GOSUB 31680:NEXT YFLD:GOTO 2200
2460 IP=0:COM$="INPUTS NOT ENTERED!!":GOSUB 32310:RETURN
2480 '
2490 'RUN SIMULATION CALCULATE DATA
2500 IF EM=0 THEN COM$="ENTER MODEL FIRST!!":GOSUB 32310:RETURN
2510 IF IP=0 THEN COM$="ENTER INPUTS FIRST!!":GOSUB 32310:RETURN
2540 RS=1:CLS:LOCATE 11,29,0:COLOR 3,0:PRINT"CALCULATING DATA.....
"
2550 IF ARC%=1 THEN GOSUB 31710:COM$="-RUN SIMULATION-":GOSUB 30410
2555 GOSUB 31740:IF RS=0 THEN RETURN
2560 FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(2):SW(I,J)=0
2570 FOR K%=1 TO KDIM(1):SW(I,J)=SW(I,J)+SP1(I,K%)*B(K%,J)
2580 NEXT K%,J,I
2590 N%=0:NN%=0:NNN%=0:KDIM(4)=(KDIM(5)/T+1):IF KDIM(4)<=1000 THEN
KDIM(6)=1:GOTO 2600
2595 KDIM(6)=INT(KDIM(4)/1000+.6):KDIM(4)=INT(KDIM(4)/KDIM(6))
2596 IF KDIM(4)>1000 THEN KDIM(6)=KDIM(6)+1:KDIM(4)=INT((KDIM(5)/T+
1)/KDIM(6)):GOTO 2596

```

```

2600 FOR I=1 TO KDIM(2): IF SU(I,1)=0 THEN SU(I,6)=0: GOTO 2650
2610 IF T*N%<SU(I,4) OR T*N%>SU(I,5) THEN SU(I,6)=0: GOTO 2650
2620 IF SU(I,1)=1 THEN SU(I,6)=SU(I,2): GOTO 2650
2630 IF SU(I,1)=2 THEN SU(I,6)=SU(I,2)+SU(I,3)*T*N%: GOTO 2650
2640 IF SU(I,1)=3 THEN SU(I,6)=SU(I,2)*SIN(N%*T*SU(I,3))
2650 NEXT I
2660 FOR I=1 TO KDIM(3): SDI(I,1)=0: SG(I,1)=0
2670 FOR J=1 TO KDIM(1): SG(I,1)=SG(I,1)+C(I,J)*SX(J): NEXT J
2680 FOR J=1 TO KDIM(2): SDI(I,1)=SDI(I,1)+D(I,J)*SU(J,6): NEXT J
2690 SP1(I,1)=SG(I,1)+SDI(I,1): NEXT I
2692 IF NN%=KDIM(6) OR NN%=0 THEN NN%=0 ELSE 2720
2694 NNN%=NNN%+1
2700 OP(NNN%,1)=N%*T: FOR K%=1 TO KDIM(3)
2710 OP(NNN%,K%+1)=SP1(K%,1): NEXT K%
2720 IF N%*T>=KDIM(5) THEN RETURN
2730 N%=N%+1: NN%=NN%+1
2740 FOR I=1 TO KDIM(1): SG(I,1)=0: SDI(I,1)=0
2750 FOR J=1 TO KDIM(1): SG(I,1)=SG(I,1)+SP0(I,J)*SX(J): NEXT J
2760 FOR J=1 TO KDIM(2): SDI(I,1)=SDI(I,1)+SW(I,J)*SU(J,6): NEXT J
2770 SX(I)=SG(I,1)+SDI(I,1): NEXT I
2780 GOTO 2600
2800 '
2810 ' GRAPHING ROUTINE
2820 RG=0: IF ARC%=1 THEN GOSUB 31710: COM$="-GRAPH DATA-": GOSUB 3041
0
2830 SCREEN 0,1: CLS: COLOR 14,0: LOCATE 3,36,0: PRINT"OPTIONS": COLOR 1
0,0
2840 LOCATE 5,24,0: PRINT"<Ctrl-F1>....GRAPH CURRENT DATA"
2850 LOCATE 6,24,0: PRINT"<Ctrl-F2>....RE-DISPLAY SCREEN FILE"
2860 LOCATE 7,24,0: PRINT"<ESC>.....END"
2870 COLOR 14,0: LOCATE 12,29,0: PRINT"WHILE GRAPH DISPLAYED": COLOR 1
0,0
2880 LOCATE 14,24,0: PRINT"<ENTER>.....CURSOR COORDINATES"
2900 LOCATE 15,24,0: PRINT"<Ctrl-F4>....SAVE GRAPH ON DISK"
2910 LOCATE 16,24,0: PRINT"<Ctrl-F5>....ZOOM"
2920 LOCATE 17,24,0: PRINT"<ESC>.....END": COLOR 14,0
2930 LOCATE 19,29,0: PRINT"(EXCEPT ON RE-DISPLAY)"
2940 INFLG=2: GOSUB 7120: IF ESCK=1 THEN RETURN
2950 IF C=94 THEN GOSUB 3000: GOTO 2830
2960 IF C=95 THEN GOSUB 3400: GOTO 2830
2970 GOSUB 30760: GOTO 2940
2980 '
2990 ' GRAPH CURRENT DATA
3000 IF RS=0 THEN COM$="NO DATA TO GRAPH!!": GOSUB 32310: RETURN
3030 CLS: IF ARC%=1 THEN COM$="Graphed current data": GOSUB 30410
3040 FOR I=1 TO KDIM(3): SDI(I,1)=0: NEXT I
3050 IF KDIM(3)=1 THEN SDI(1,1)=1: ABSC%=1: GOTO 3190
3060 COLOR 12,0: LOCATE 9,27,0: PRINT"OUTPUT# FOR ABSCISSA >"SPC(12)
3070 IF HELP%=1 THEN LOCATE 10,30,0: PRINT"(Default = Time)"
3080 NUMDIG=10: INFLG=3: XFLD=51: YFLD=9: GOSUB 7110: IF ESCK=1 THEN RET
URN

```

```

3090 IF S<0 OR S>KDIM(3) THEN GOSUB 30760:GOTO 3060
3100 ABSC%=S+1: IF ABSC%=1 THEN LOCATE 9,51,0:PRINT"TIME";
3110 COLOR 12,0:LOCATE 12,27,0:PRINT"OUTPUTS TO BE GRAPHED >"SPC(12
)
3120 IF HELP%=1 THEN LOCATE 13,30,0:PRINT"(Default = All)"
3130 YFLD=12:XFLD=51:COLOR 6,0:FOR I=1 TO KDIM(3)
3140 GOSUB 7110:S=INT(S)
3150 IF S=0 AND I=1 THEN FOR J=1 TO KDIM(3):SDI(J,1)=1:NEXT J:LOCAT
E 12,51,0:PRINT"ALL";:GOTO 3190
3160 IF S=0 THEN 3190 ELSE IF S<1 OR S>KDIM(3) THEN GOSUB 3370:GOTO
3140
3170 IF SDI(S,1)=1 OR S=ABSC%-1 THEN GOSUB 3370:GOTO 3140
3180 SDI(S,1)=1:XFLD=SXFLD:LOCATE YFLD,XFLD,0:PRINT S:XFLD=XFLD+4:N
EXT I
3190 COLOR 12,0:LOCATE 15,27,0:PRINT"GRIDS ON GRAPH? (Y/N) >"
3200 IF HELP%=1 THEN LOCATE 16,30,0:PRINT"(Default = No)"
3210 INFLG=1:GOSUB 7110:LOCATE 15,51,0:IF C$="N" THEN PRINT"NO":PGD
=0:GOTO 3230
3220 PRINT"YES":PGD=1
3230 PN=KDIM(4):PMINX=0:PMAXY=0:PMINY=0:PFM=1
3240 IF ABSC%=1 THEN PMAXX=KDIM(5) ELSE PMAXX=0
3250 FOR I=1 TO KDIM(4):PX(I)=OP(I,ABSC%):IF ABSC%=1 THEN 3280
3260 IF OP(I,ABSC%)<PMINX THEN PMINX=OP(I,ABSC%)
3270 IF OP(I,ABSC%)>PMAXX THEN PMAXX=OP(I,ABSC%)
3280 FOR J=2 TO KDIM(3)+1:IF J=ABSC% OR SDI(J-1,1)=0 THEN 3310
3290 IF OP(I,J)<PMINY THEN PMINY=OP(I,J)
3300 IF OP(I,J)>PMAXY THEN PMAXY=OP(I,J)
3310 NEXT J,I
3320 PFLG=1:GOSUB 4620:IF P=0 THEN RETURN
3330 PFLG=2:FOR I=2 TO KDIM(3)+1:IF I=ABSC%-1 OR SDI(I-1,1)=0 THEN
3350
3340 FOR J=1 TO KDIM(4):PY(J)=OP(J,I):NEXT J:GOSUB 4620
3350 NEXT I:GOSUB 6020:IF ZOOM=0 THEN RETURN ELSE PFLG=1:GOSUB 4620
3360 IF P=0 THEN RETURN ELSE GOTO 3330
3370 GOSUB 30760:XFLD=SXFLD:LOCATE YFLD,XFLD,0:PRINT SPC(12):RETURN
3380 '
3390 'RE-GRAPH SCREEN FILE
3400 CLS:FILETPE=1:IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
3410 FILEXT$=".GPH":FILENUM=1:GOSUB 30650
3420 IF ESCK=1 OR FERR$<>"OK" THEN RETURN ELSE CLOSE #FILENUM:RG=1
3430 CLS:SCREEN 2:DEF SEG=&HB800
3440 ON ERROR GOTO 3510:BLOAD FILENAM$,0:DEF SEG:ON ERROR GOTO 0
3450 IF ARC%=1 THEN COM$="Re-graph from screen file "+FILENAM$:GOSU
B 30410
3470 INFLG=2:GOSUB 7120:IF ESCK=1 THEN RETURN
3500 GOSUB 30760:GOTO 3470
3510 ON ERROR GOTO 0:CLS:SCREEN 0,1:COLOR 15,0:GOSUB 30760:LOCATE 1
0,30,0
3520 PRINT"ERROR GRAPHING FILE":COLOR 7,0:LOCATE 12,33,0:PRINT RT$:
INFLG=1
3530 GOSUB 7120:IF C$="Y" AND ESCK=0 THEN RESUME 3400 ELSE RESUME 3

```

```

540
3540 IF ARC%=1 THEN COM$="ERROR REGRAPHING FILE "+FILENAM$:GOSUB 30
410
3560 RETURN
3570 '
3580 'SAVE/LIST DATA
3590 IF ARC%=1 THEN GOSUB 31710:COM$="-SAVE/LIST DATA-":GOSUB 30410
3600 CLS:COLOR 14,0:LOCATE 4,36,0:PRINT"OPTIONS":COLOR 10,0
3610 LOCATE 6,25,0:PRINT"<Ctrl-F6>....LIST CURRENT DATA"
3620 LOCATE 7,25,0:PRINT"<Ctrl-F7>....SAVE DATA ON DISK"
3630 LOCATE 8,25,0:PRINT"<Ctrl-F8>....RECALL DATA FILE"
3640 LOCATE 9,25,0:PRINT"<ESC>.....END"
3650 INFLG=2:GOSUB 7120:IF ESCK=1 THEN RETURN
3660 IF C=99 THEN GOSUB 3720:GOTO 3600
3670 IF C=100 THEN GOSUB 3890:GOTO 3600
3680 IF C=101 THEN GOSUB 4070:GOTO 3600
3690 GOSUB 30760:GOTO 3650
3700 '
3710 'LIST CURRENT DATA
3720 IF RS=0 THEN COM$="NO DATA TO LIST!!":GOSUB 32310:RETURN
3740 CLS:COL1=2:IF ARC%=1 THEN COM$="Listed current data":GOSUB 304
10
3750 IF (KDIM(3)+1)/4<=1 THEN COL2=KDIM(3)+1:GOSUB 3780:RETURN
3760 COL2=4:GOSUB 3780:COL1=5:COL2=KDIM(3)+1:GOSUB 3780:RETURN
3770 '
3780 YFLD=3:XFLD=19:LCD=21:CLS:COLOR 6,0
3790 FOR I=1 TO KDIM(4):IF I/LCD<1 THEN 3810 ELSE LCD=LCD+21:GOSUB
3850
3800 IF ESCK=1 THEN RETURN
3810 LOCATE YFLD,1,0:PRINT OP(I,1):FOR J=COL1 TO COL2
3820 LOCATE YFLD,XFLD,0:PRINT OP(I,J);:XFLD=XFLD+22
3830 NEXT J:YFLD=YFLD+1:XFLD=19:NEXT I:GOSUB 3850:RETURN
3840 '
3850 COLOR 12,0:LOCATE 1,1,0:PRINT"TIME(SEC) ";:FOR K%=COL1 TO COL2
:XFLD=28:YFLD=25
3860 PRINT SPC(10)"OUTPUT #"+K%-1;:NEXT K%:GOSUB 30800:CLS:XFLD=19:Y
FLD=3:COLOR 6,0:RETURN
3870 '
3880 'SAVE DATA ON DISK
3890 IF RS=0 THEN COM$="NO DATA TO SAVE!!":GOSUB 32310:RETURN
3910 CLS:FILETPE=0:IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
3920 FILEXT$=".SDT":FILENUM=1:GOSUB 30650
3930 IF ESCK=0 AND FERR$="OK" THEN 3960
3940 COM$="DATA NOT SAVED!!":GOSUB 32310:RETURN
3960 SFILE$=FILENAM$:CLS:COLOR 3,0:LOCATE 11,31,0:PRINT"SAVING DATA
....."
3970 MAT=0:ROW%=5:COL%=1:GOSUB 31380:IF FERR$<>"OK" THEN 4010
3980 MAT=6:ROW%=KDIM(4):COL%=KDIM(3)+1:GOSUB 31380:IF FERR$<>"OK" T
HEN 4010 ELSE CLOSE #FILENUM
3990 IF ARC%=1 THEN COM$="Data saved in file "+SFILE$:GOSUB 30410
4000 RETURN

```

```

4010 CLOSE #FILENUM:CLS:GOSUB 30760:COLOR 15,0:LOCATE 10,28,0:PRINT
      FERR$
4020 COLOR 7,0:LOCATE 12,33:PRINT RT$:INFLG=1:GOSUB 7120:IF C$="Y"
AND ESCK=0 THEN 3910
4030 IF ARC%=1 THEN COM$=FERR$+" while saving data":GOSUB 30410
4040 FILENAM$=SFILE$:GOSUB 31530:GOTO 3940
4050 '
4060 'RECALL DATA FILE
4070 IF RS=1 THEN COM$="ERASE EXISTING DATA? (Y/N)":GOSUB 32340:IF
      C$="N" OR ESCK=1 THEN RETURN
4100 CLS:FILETPE=1:IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
4110 FILEXT$=".SDT":FILENUM=1:GOSUB 30650
4120 IF ESCK=0 AND FERR$="OK" THEN SFILE$=FILENAM$:GOTO 4160
4130 COM$="DATA NOT RECALLED!!":GOSUB 32310:RETURN
4160 CLS:COLOR 3,0:LOCATE 10,29,0:PRINT"RECALLING DATA....."
4170 ROW%=5:COL%=1:MAT=0:GOSUB 31110:IF FERR$<>"OK" THEN 4210
4180 ROW%=KDIM(4):COL%=KDIM(3)+1:MAT=6:GOSUB 31110:IF FERR$<>"OK" T
HEN 4210
4190 RS=1:CLOSE #FILENUM:IF ARC%=1 THEN COM$="Recalled data from fi
      le "+SFILE$:GOSUB 30410
4200 RETURN
4210 CLOSE #FILENUM:GOSUB 30760:CLS:COLOR 15,0:LOCATE 10,28,0:PRINT
      FERR$
4220 COLOR 7,0:LOCATE 12,33,0:PRINT RT$:INFLG=1:GOSUB 7120
4230 IF C$="Y" AND ESCK=0 THEN 4100 ELSE RS=0:GOTO 4130
4240 '
4250 'MODIFY CURRENT MODEL
4260 IF EM=0 THEN COM$="NO MODEL TO MODIFY!!":GOSUB 32310:RETURN
4280 IF ARC%=1 THEN GOSUB 31710:COM$="-MODIFY CURRENT MODEL-":GOSUB
      30410
4290 CLS:IF HELP%=0 THEN 4310 ELSE LOCATE 1,4,0:COLOR 3,0
4300 PRINT"ENTER LETTER OF MATRIX TO CHANGE, S FOR SYSTEM DIMENSION
      S, ESC TO QUIT."
4310 COLOR 12,0:LOCATE 3,24,0:PRINT"(A, B, C, D, X, S, OR ESC) >":I
NFLG=5:NUMDIG=1:YFLD=3:XFLD=53:GOSUB 7110
4320 IF ESCK=1 THEN RETURN
4330 IF S$="S" OR S$="s" THEN GOSUB 4400:GOTO 4290
4340 IF S$="A" OR S$="a" THEN COM$="A MATRIX":MAT=1:ROW%=KDIM(1):CO
      L%=KDIM(1):GOSUB 32440:GOTO 4290
4350 IF S$="B" OR S$="b" THEN COM$="B MATRIX":MAT=2:ROW%=KDIM(1):CO
      L%=KDIM(2):GOSUB 32440:GOTO 4290
4360 IF S$="C" OR S$="c" THEN COM$="C MATRIX":MAT=3:ROW%=KDIM(3):CO
      L%=KDIM(1):GOSUB 32440:GOTO 4290
4370 IF S$="D" OR S$="d" THEN COM$="D MATRIX":MAT=4:ROW%=KDIM(3):CO
      L%=KDIM(2):GOSUB 32440:GOTO 4290
4380 IF S$="X" OR S$="x" THEN COM$="X0 MATRIX":MAT=5:ROW%=KDIM(1):C
      OL%=1:GOSUB 32440:GOTO 4290
4390 GOSUB 30760:GOTO 4310
4400 CLS:LOCATE 1,1,0:COLOR 15,0:PRINT"Changing dimensions without
      changing matrices can lead to unexpected results!!"
4410 COLOR 12,0:LOCATE 3,1,0:PRINT"SYSTEM DIMENSION >"SPC(8)"NUMBER

```

```

OF INPUTS >"SPC(8)"NUMBER OUTPUTS >"
4420 COLOR 6,0:LOCATE 3,20,0:PRINT KDIM(1):LOCATE 3,46,0:PRINT KDIM
(2):LOCATE 3,72,0:PRINT KDIM(3)
4430 GOSUB 1030:IF ESCK=1 OR C$="N" THEN RETURN ELSE 4400
4570 '
4580 ' PLOTTING SUBROUTINE
4590 ' PFLG=0 -axes & plot : PFLG=1 -axes only : PLFG=2 -plot only
4600 ' PGD=1 grids on plot : PGD=0 no grids : PFM=1 don't find min/m
ax
4610 ' Plots PX() and PY() arrays where PN = number of xy pairs
4620 IF PFLG>1 THEN GOSUB 5490:RETURN
4630 GOSUB 4720
4640 IF PFM=1 THEN GOSUB 5170:IF P=0 THEN RETURN ELSE GOSUB 4680:GO
SUB 5240:GOTO 4660
4650 GOSUB 4820:GOSUB 5170:IF P=0 THEN RETURN ELSE GOSUB 4680:GOSUB
5240
4660 IF PFLG=1 THEN RETURN ELSE GOSUB 5450
4670 RETURN
4680 ' TITLE
4690 LOCATE 1,7
4700 PRINT"SCALE X-AXIS BY";STR$(PSCLX);", Y-AXIS BY";PSCLY;
4710 RETURN
4720 ' INITIALIZATION
4730 DEF FNPSC(PS,PXS,PNS,PXO,PNO)=INT(PXO-(PXO-PNO)/(PXS-PNS)*(PXS
-PS)+.5)
4750 PNINX=6.5:PNINY=5.5:SCREEN 2:CLS:PN=PN-1
4760 PLNL=50:PLNR=589:PLNB=173:PLNT=12:PLOCX1=8
4770 PLOCY1=INT(PLNT/8+.5):PLOCY2=PLOCY1+1
4780 PLOCY3=INT(PLNB/8+.625):PLOCY4=PLOCY3+1
4790 PLOCY5=PLOCY4+1:PLOCX2=INT(PLNL/8-4)
4800 PLOCX3=PLOCX2+1:PLOCX4=PLOCX3+INT((PLNR-PLNL)/8+.625)-1
4810 RETURN
4820 ' FIND MINIMUM & MAXIMUM OF ARRAYS
4830 PMINX=PX(1):PMAXX=PMINX:PMINY=PY(1):PMAXY=PMINY
4840 FOR PII%=2 TO PN
4850 IF PMINX>PX(PII%) THEN PMINX=PX(PII%)
4860 IF PMAXX<PX(PII%) THEN PMAXX=PX(PII%)
4870 IF PMINY>PY(PII%) THEN PMINY=PY(PII%)
4880 IF PMAXY<PY(PII%) THEN PMAXY=PY(PII%)
4890 NEXT PII%
4900 RETURN
4910 ' FIND LINEAR SCALE INTERVALS
4920 IF PMIN=PMAX THEN PMIN=PMIN-1:PMAX=PMAX+1
4950 P=1:PEPS=.025:PA=ABS(PMIN)
4960 IF ABS(PMIN)<ABS(PMAX) THEN PA=ABS(PMAX)
4970 PSCL=10*INT(LOG(PA)/LOG(10))
4980 PMINA=PMIN/PSCL:PMAXA=PMAX/PSCL
4990 PD=(PMAXA-PMINA)/PNIN:PJ=PD*PEPS:PE=INT(LOG(PD)/LOG(10))
5000 PF=PD/10*PE:PV=10
5010 IF PF<SQR(50) THEN PV=5
5020 IF PF<SQR(10) THEN PV=2

```

```

5030 IF PF<SQR(2) THEN PV=1
5040 PWDT=PV*100PE: PG=INT(PMINA/PWDT)
5050 IF ABS(PG+1-PMINA/PWDT)<PJ THEN PG=PG+1
5060 PMINP=PWDT*PG: PH=INT(PMAXA/PWDT)+1
5070 IF ABS(PMAXA/PWDT+1-PH)<PJ THEN PH=PH-1
5080 PMAXP=PWDT*PH: PNINP=PH-PG
5090 IF ABS(PMAXP)>=10 OR ABS(PMINP)>=10 THEN PSCL=PSCL*10: GOTO 498
0
5100 RETURN
5110 ' LINEAR TRANSFORMATION FROM SUBJECT TO OBJECT SPACE
5120 PX1=FNPSC(PX(P11%-1)/PSCLX,PMAXPX,PMINPX,PLNR,PLNL)
5130 PX2=FNPSC(PX(P11%)/PSCLX,PMAXPX,PMINPX,PLNR,PLNL)
5140 PY1=FNPSC(PY(P11%-1)/PSCLY,PMAXPY,PMINPY,PLNT,PLNB)
5150 PY2=FNPSC(PY(P11%)/PSCLY,PMAXPY,PMINPY,PLNT,PLNB)
5160 RETURN
5170 ' FIND X & Y LINEAR SCALE INTERVALS
5180 ' X SCALE
5190 PMIN=PMINX: PMAX=PMAXX: PNIN=PNINX: GOSUB 4910: IF P=0 THEN RETURN
5200 PSCLX=PSCL: PMINPX=PMINP: PMAXPX=PMAXP: PNINPX=PNINP: PWDTX=PWDT
5210 ' Y SCALE
5220 PMIN=PMINY: PMAX=PMAXY: PNIN=PNINY: GOSUB 4910: IF P=0 THEN RETURN
5230 PSCLY=PSCL: PMINPY=PMINP: PMAXPY=PMAXP: PNINPY=PNINP: PWDY=PWDT: R
ETURN
5240 ' DRAW LINEAR GRIDS & LABELS
5250 PLOCY=(PLOCY3-PLOCY2+1)/PNINPY: PLNWX=(PLNB-PLNT)/PNINPY: X1=PL
NL-49: X2=PLNL-4
5260 ' Y GRID & LABELS
5270 FOR P11=0 TO PNINPY: PLNY1=PLNB-P11*PLNWX
5280 Y1=PLNY1-5: Y2=PLNY1+5: S$=STR$(PMINPY+P11*PWDY): GOSUB 6850
5290 IF PGD=1 THEN 5320
5300 IF P11=0 OR P11=PNINPY THEN 5320
5310 LINE(PLNL-4,PLNY1)-(PLNL,PLNY1): LINE(PLNR,PLNY1)-(PLNR+4,PLNY1
): GOTO 5330
5320 LINE(PLNL,PLNY1)-(PLNR,PLNY1)
5330 NEXT P11
5340 PLOCX=(PLOCX4-PLOCX3+1)/PNINPX: PLNWX=(PLNR-PLNL)/PNINPX
5350 ' X GRID & LABELS
5360 FOR P11=0 TO PNINPX
5370 IF P11/2=INT(P11/2) THEN PLOC=PLOCY5 ELSE PLOC=PLOCY4
5380 PLNX1=PLNL+P11*PLNWX: Y1=PLOC*8-8: Y2=Y1+8: X1=PLNX1-30: X2=PLNX1+
25
5390 S$=STR$(PMINPX+P11*PWDX): GOSUB 6850: IF PGD=1 THEN 5420
5400 IF P11=0 OR P11=PNINPX THEN 5420
5410 LINE(PLNX1,PLNB)-(PLNX1,PLNB+2): LINE(PLNX1,PLNT-2)-(PLNX1,PLNT
): GOTO 5430
5420 LINE(PLNX1,PLNB)-(PLNX1,PLNT)
5430 NEXT P11
5440 RETURN
5450 ' DRAW X-Y CURVE
5460 FOR P11%=2 TO PN: GOSUB 5110
5470 IF P11%>2 THEN LINE -(PX2,PY2) ELSE LINE(PX1,PY1)-(PX2,PY2)

```

```

5480 NEXT PII%: RETURN
5490 'CLIPPING ROUTINE TO CHECK FOR POINTS OUTSIDE GRID
5500 FOR PII%=2 TO PN: GOSUB 5110
5510 IF PX1<PLNL OR PX1>PLNR OR PY1>PLNB OR PY1<PLNT THEN PFL1=1 ELSE PFL1=0
5520 IF PX2<PLNL OR PX2>PLNR OR PY2>PLNB OR PY2<PLNT THEN PFL2=1 ELSE PFL2=0
5530 IF PFL1=0 AND PFL2=0 THEN 5960
5540 IF PFL1*PFL2=1 THEN 5760
5550 'ONE POINT OUTSIDE, ONE POINT INSIDE
5560 IF PX1=PX2 THEN 5660
5570 IF PFL1=1 AND PX1>PLNR OR PFL2=1 AND PX2>PLNR THEN 5620
5580 PYT=FNPS(PLNL,PX1,PX2,PY1,PY2)
5590 IF PYT>PLNB OR PYT<PLNT THEN 5620
5600 IF PFL2=1 THEN PX2=PLNL: PY2=PYT: GOTO 5960
5610 PX1=PLNL: PY1=PYT: GOTO 5960
5620 PYT=FNPS(PLNR,PX1,PX2,PY1,PY2)
5630 IF PYT>PLNB OR PYT<PLNT THEN 5660
5640 IF PFL2=1 THEN PX2=PLNR: PY2=PYT: GOTO 5960
5650 PX1=PLNR: PY1=PYT: GOTO 5960
5660 IF PY1=PY2 THEN 5760
5670 IF PFL1=1 AND PY1<PLNT OR PFL2=1 AND PY2<PLNT THEN 5720
5680 PXT=FNPS(PLNB,PY1,PY2,PX1,PX2)
5690 IF PXT<PLNL OR PXT>PLNR THEN 5720
5700 IF PFL2=1 THEN PY2=PLNB: PX2=PXT: GOTO 5960
5710 PY1=PLNB: PX1=PXT: GOTO 5960
5720 PXT=FNPS(PLNT,PY1,PY2,PX1,PX2)
5730 IF PFL2=1 THEN PY2=PLNT: PX2=PXT: GOTO 5960
5740 PY1=PLNT: PX1=PXT: GOTO 5960
5750 'BOTH POINTS LIE OUTSIDE
5760 PTEMP=0: IF PX1<PLNL AND PX2<PLNL OR PX1>PLNR AND PX2>PLNR THEN
5770 5970
5770 IF PY1<PLNT AND PY2<PLNT OR PY1>PLNB AND PY2>PLNB THEN 5970
5780 IF PX1=PX2 THEN 5870
5790 PYT=FNPS(PLNL,PX1,PX2,PY1,PY2)
5800 IF PYT>PLNB OR PYT<PLNT THEN 5820
5810 PX1=PLNL: PY1=PYT: PTEMP=1
5820 PYT=FNPS(PLNR,PX1,PX2,PY1,PY2)
5830 IF PYT>PLNB OR PYT<PLNT THEN 5870
5840 PTEMP=PTEMP+1
5850 IF PTEMP=2 THEN PX2=PLNR: PY2=PYT: GOTO 5960
5860 PX1=PLNR: PY1=PYT
5870 IF PY1=PY2 THEN 5970
5880 PXT=FNPS(PLNB,PY1,PY2,PX1,PX2)
5890 IF PXT<PLNL OR PXT>PLNR THEN 5930
5900 PTEMP=PTEMP+1
5910 IF PTEMP=2 THEN PY2=PLNB: PX2=PXT: GOTO 5960
5920 PY1=PLNB: PX1=PXT
5930 IF PTEMP=0 THEN 5970
5940 PXT=FNPS(PLNT,PY1,PY2,PX1,PX2)
5950 PY2=PLNT: PX2=PXT

```

```

5960 LINE(PX1,PY1)-(PX2,PY2)
5970 NEXT PII%
5980 RETURN
5990 '
6000 'CURSOR SUBROUTINE
6010 'Moves cursor around screen keeping pixel coordinates
6020 DEF SEG=0:POKE 1047,64:DEF SEG:IF SND%>1 THEN BEEP
6030 CRSRX=PLNL:CRSRY=PLNB:ZOOM=0:GOSUB 6830
6040 IF CRSRX<PLNL THEN CRSRX=PLNL
6050 IF CRSRX>PLNR THEN CRSRX=PLNR
6060 IF CRSRY<PLNT THEN CRSRY=PLNT
6070 IF CRSRY>PLNB THEN CRSRY=PLNB
6080 GOSUB 6300:GOSUB 30030 'Draw csr & clear KBD buf
6090 C$=INKEY$:IF C$="" THEN 6090
6100 IF C$=CHR$(27) THEN ZOOM=0:RETURN
6110 IF C$=CHR$(13) THEN GOSUB 6470 'Wants coordinates
6120 IF LEN(C$)=2 THEN C$=RIGHT$(C$,1) ELSE 6230
6150 IF C$=CHR$(98) THEN GOSUB 6550:IF ESCK<>1 THEN ZOOM=1:RETURN
6160 IF C$=CHR$(97) THEN GOSUB 6670:GOSUB 6830
6170 GOSUB 6380 'Erase CRSR
6180 IF C$=CHR$(72) THEN CRSRY=CRSRY-1:GOTO 6040
6190 IF C$=CHR$(80) THEN CRSRY=CRSRY+1:GOTO 6040
6200 IF C$=CHR$(77) THEN CRSRX=CRSRX+1:GOTO 6040
6210 IF C$=CHR$(75) THEN CRSRX=CRSRX-1:GOTO 6040
6220 DEF SEG=0:POKE 1047,64:DEF SEG:GOTO 6080
6230 GOSUB 6380 'Erase CRSR
6240 IF C$=CHR$(56) THEN CRSRY=CRSRY-5:GOTO 6040
6250 IF C$=CHR$(50) THEN CRSRY=CRSRY+5:GOTO 6040
6260 IF C$=CHR$(54) THEN CRSRX=CRSRX+10:GOTO 6040
6270 IF C$=CHR$(52) THEN CRSRX=CRSRX-10:GOTO 6040
6280 DEF SEG=0:POKE 1047,64:DEF SEG:GOTO 6080
6290 'DRAW CURSOR
6300 CRSRPX=CRSRX-6:CRSRPY=CRSRY-6
6310 FOR CII%=1 TO 11
6320 CRSRPX=CRSRPX+1:CRSRPY=CRSRPY+1
6330 SW(CII%,1)=POINT(CRSRPX,CRSRY):SW(CII%,2)=POINT(CRSRX,CRSRPY)
6340 PSET(CRSRPX,CRSRY):PSET(CRSRX,CRSRPY)
6350 NEXT CII%
6360 RETURN
6370 'ERASE CURSOR
6380 CRSREX=CRSRX-6:CRSREY=CRSRY-6
6390 FOR CII%=1 TO 11
6400 CRSREX=CRSREX+1:CRSREY=CRSREY+1
6410 PRESET(CRSREX,CRSRY):PRESET(CRSRX,CRSREY)
6420 IF SW(CII%,1)=1 THEN PSET(CRSREX,CRSRY)
6430 IF SW(CII%,2)=1 THEN PSET(CRSRX,CRSREY)
6440 NEXT CII%
6450 RETURN
6460 'LINEAR COORDINATE FINDER
6470 PBY=PMINPY:PTY=PMINPY+PNINPY*PWDY
6480 PLX=PMINPX:PRX=PMINPX+PNINPX*PWTX

```

```

6490 PXCOD=(PRX-(PLNR-CRSRX)*(PRX-PLX)/(PLNR-PLNL))*PSCLX
6500 PYCOD=(PBY-(PLNB-CRSRY)*(PBY-PTY)/(PLNB-PLNT))*PSCLY
6510 LOCATE 1,47:PRINT USING "X= ###.###cccc";PXCOD
6520 LOCATE 1,64:PRINT USING "Y= ###.###cccc";PYCOD:RETURN
6530 '
6540 'ZOOM ROUTINE
6550 LOCATE 1,47,0:PRINT"X1="SPC(30):XFLD=51:YFLD=1:
6560 NUMDIG=10:INFLG=3:GOSUB 7120:IF ESCK=1 THEN 6640 ELSE TPMINX=S
6570 LOCATE 1,62,0:PRINT"X2="SPC(14):XFLD=66:YFLD=1:GOSUB 7120:IF E
SCK=1 THEN 6640 ELSE TPMAXX=S
6580 IF TPMAXX<=TPMINX THEN GOSUB 30760:GOTO 6570
6590 LOCATE 1,47,0:PRINT"Y1="SPC(29):XFLD=51:YFLD=1
6600 GOSUB 7120:IF ESCK=1 THEN 6640 ELSE TPMINY=S
6610 LOCATE 1,62,0:PRINT"Y2="SPC(14):XFLD=66:YFLD=1:GOSUB 7120:IF E
SCK=1 THEN 6640 ELSE TPMAXY=S
6620 IF TPMAXY<=TPMINY THEN GOSUB 30760:GOTO 6610
6630 PMINX=TPMINX:PMAXX=TPMAXX:PMINY=TPMINY:PMAXY=TPMAXY:RETURN
6640 LOCATE 1,47,0:PRINT SPC(30):RETURN
6650 '
6660 'SCREEN DUMP TO DISK
6670 LOCATE 1,47,0:PRINT"FILENAME >"SPC(20):NUMDIG=10:YFLD=1:XFLD=5
8
6680 INFLG=4:GOSUB 7120:IF ESCK=1 THEN 6800 ELSE FILENUM=1:FILETPE=
0
6690 FILENAM$=S$+".GPH":GOSUB 30090
6700 IF FERR$="OK" THEN CLOSE #FILENUM:LOCATE 1,47,0:PRINT SPC(30):
GOTO 6750
6710 LOCATE 1,47,0:PRINT FERR$ SPC(1) RT$:GOSUB 30760:INFLG=1:GOSUB
7120
6720 IF C$="Y" AND ESCK=0 THEN 6670 ELSE LOCATE 1,47,0:PRINT SPC(30
)
6730 IF ARC%=1 THEN COM$=FERR$+" on screen dump to disk":GOSUB 3041
0
6740 GOSUB 31530:RETURN
6750 ON ERROR GOTO 6790:DEF SEG=&HB800:BSAVE FILENAM$,0,&H4000:DEF
SEG
6770 IF ARC%=1 THEN COM$="Saved Graphics Screen in file "+FILENAM$:
GOSUB 30410
6780 ON ERROR GOTO 0:RETURN
6790 ON ERROR GOTO 0:GOSUB 31260:RESUME 6710
6800 LOCATE 1,47,0:PRINT SPC(30):RETURN
6810 '
6820 'TITLE ON GRAPH
6830 LOCATE 25,34,0:IF ABSC%=1 THEN PRINT"    TIME    ";ELSE PRINT"O
UTPUT #";ABSC%-1;
6835 S$='OUTPUT':FOR I=1 TO 6:T$=LEFT$(S$,1):S$=MID$(S$,2)
6840 LOCATE I+8,1,0:PRINT T$:NEXT I:RETURN
6845 '
6850 'PLOT GRAPH LABELS ROUTINE
6860 'Allows precise placement of a string in graphics mode
6870 '(X1,Y),(X2,Y) are graphics coordinates

```

```

6880 'S$ = string to be placed
6890 'string centered horizontally in the field
6900 LS=LEN(S$):C=INT((X2-X1)/8):IF LS>C THEN S$=LEFT$(S$,C)
6910 X=X1+(X2-X1+1)/2-8*LEN(S$)/2+4:Y=Y1+(Y2-Y1+1)/2-4
6920 WHILE S$<>""
6930 IF X>639 THEN 6950 ELSE C$=LEFT$(S$,1):S$=MID$(S$,2)
6940 GOSUB 6960:IF X>=0 THEN X=X+8
6950 WEND:RETURN
6960 'PLACE CHARACTER AT PARTICULAR COORDINATES
6970 A%(1)=8:A%(2)=8:DEF SEG=&HF000:OFFSET=8*ASC(C$)+&HFA6E
6980 FOR J=1 TO 8:B$(J)=HEX$(PEEK(OFFSET+J-1))
6990 IF LEN(B$(J))=1 THEN B$(J)='0'+B$(J)
7000 NEXT J
7010 FOR J=1 TO 4:A%(J+2)=VAL("&H"+B$(2*J)+B$(2*J-1)):NEXT J
7020 DEF SEG:PUT(X,Y),A%:RETURN
7030 '
7040 'INPUT ROUTINE for numbers, names, function keys, comments, &
Y/N answers
7050 'INFLG=1: gets Y/N answer, returns c$='Y' or c$='N'
7060 'INFLG=2: gets function key, returns c=asc(2nd char of key)
7070 'INFLG=3: gets number as string, checks if valid, returns s=n
umber
7080 'INFLG=4: gets filename, returns s$=filename
7090 'INFLG=5: gets comment from user for archive, returns s$=comm
ent
7100 'Calling routine specifies NUMDIG, XFLD, YFLD, and INFLG
7110 COLOR 6,0
7120 SXFLD=XFLD:S$='':C$='':T$='':N$='':ESCK=0:GOSUB 30030:IF SND%>
1 THEN BEEP
7130 IF INFLG>2 THEN 7150
7140 C$=INKEY$:IF C$="" THEN 7140 ELSE 7170
7150 LOCATE YFLD,XFLD,1
7160 C$=INKEY$:IF C$="" THEN 7150
7170 IF C$=CHR$(27) THEN ESCK=1:RETURN 'escape key
7180 IF INFLG=2 THEN 7370
7190 IF LEN(C$)>1 THEN 7450 'invalid
7200 C=ASC(C$)
7210 IF INFLG=1 THEN 7320
7220 IF C=8 THEN 7500
7230 IF INFLG=5 THEN 7800 'comment
7240 IF INFLG=3 THEN 7390 'number
7250 'Name Routine
7260 IF C=13 THEN RETURN
7270 IF C=46 AND NUMDIG>10 THEN 7470 'okay
7280 IF C=91 OR C=92 OR C=93 OR C=124 THEN 7450 'invalid
7290 IF C=34 OR C=39 OR C=42 OR C=43 OR C=46 OR C=47 THEN 7450 'in
valid
7300 IF (C>32 AND C<59) OR (C>63 AND C<127) THEN 7470 ELSE 7450
7310 'Yes/No routine
7320 IF C=89 OR C=121 THEN C$='Y':RETURN
7330 IF C=78 OR C=110 THEN C$='N':RETURN

```

```

7340 IF C=13 THEN C$="N":RETURN
7350 GOTO 7450
7360 'Function Key
7370 IF LEN(C$)<2 THEN 7450 ELSE C$=RIGHT$(C$,1):C=ASC(C$)
7380 IF (C>58 AND C<69) OR (C>83 AND C<114) THEN RETURN ELSE 7450
7390 'Number Routine
7400 IF C=13 THEN 7540 'carriage return
7410 IF C>47 AND C<58 THEN 7470 'okay
7420 IF C=46 OR C=43 OR C=45 OR C=32 THEN 7470 'okay
7430 IF C=69 OR C=101 THEN C$=CHR$(69):GOTO 7470 'okay
7440 'Illegal character
7450 GOSUB 30760:BEEP:GOTO 7130
7460 'Character okay
7470 IF LEN(S$)>=NUMDIG THEN 7450
7480 LOCATE YFLD,XFLD:PRINT C$;:S$=S$+C$:XFLD=XFLD+1:GOTO 7130
7490 'Backspace
7500 XFLD=XFLD-1:LOCATE YFLD,XFLD:PRINT" ";:IF XFLD<=SXFLD THEN XFLD=SXFLD
7510 LNT=LEN(S$)-1:IF LNT<0 THEN LNT=0
7520 S$=MID$(S$,1,LNT):GOTO 7130
7530 'Number Check routine
7540 NUMCK=0:DECPT=0:EXPNT=0:SIGN=0
7550 IF S$="" THEN 7770 ELSE T$=LEFT$(S$,1)
7560 IF T$="+" OR T$="-" THEN N$=T$:S$=MID$(S$,2)
7570 IF T$="." THEN N$=T$:DECPT=1:S$=MID$(S$,2)
7580 IF S$="" THEN 7790
7590 IF (T$="+" OR T$="-") AND INSTR(".",LEFT$(S$,1))=1 THEN DECPT=1:T$=LEFT$(S$,1):N$=N$+T$:S$=MID$(S$,2)
7600 IF INSTR(DIGIT$,LEFT$(S$,1))=0 THEN 7790
7610 WHILE S$<>"":T$=LEFT$(S$,1):S$=MID$(S$,2)
7620 IF T$=" " THEN 7680
7630 IF (T$="+" OR T$="-") AND RIGHT$(N$,1)<>"E" THEN NUMCK=1:GOTO 7680
7640 IF T$="." THEN IF DECPT=1 OR EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:DECPT=1
7650 IF T$="+" OR T$="-" THEN IF SIGN=1 OR EXPNT=0 THEN NUMCK=1 ELSE N$=N$+T$:SIGN=1
7660 IF T$="E" THEN IF EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:EXPNT=1:DECPT=1
7670 IF T$<> "." AND T$<> "+" AND T$<> "-" AND T$<> "E" THEN N$=N$+T$
7680 WEND:IF NUMCK=1 THEN 7790
7690 'Check for overflow
7700 D=0:N%=INSTR(N$,"E"):IF N%=0 THEN 7770
7710 IF N%=1 THEN 7790
7720 S$=LEFT$(N$,N%-1):C$=MID$(N$,N%+1):S=VAL(S$)
7730 IF S=0 THEN D=0 ELSE D=INT(LOG(ABS(S))/LOG(10))
7740 IF C$<> "" THEN D=D+VAL(C$)
7750 IF D<-37 OR D>37 THEN 7790
7760 'convert
7770 S=VAL(N$):RETURN
7780 'invalid

```

```

7790 NUMCK=1:GOSUB 30760:BEEP:LOCATE YFLD,SXFLD,0:PRINT SPC(NUMDIG)
:XFLD= SXFLD:GOTO 7120
7800 'Comment routine
7810 IF C$=CHR$(13) THEN RETURN
7820 IF LEN(S$)>=NUMDIG THEN RETURN ELSE 7480
7830 '
30000 '***** SUBROUTINES *****
30010 '
30020 'CLEAR KBD BUFFER
30030 DEF SEG=0:POKE 1050,PEEK(1052):DEF SEG
30040 RETURN
30050 '
30060 'OPEN DISK FILE FOR INPUT/OUTPUT
30070 'Calling routine specifies filename$, filename, fileapp, & file
tpe
30080 'fileapp=0 no append, fileapp=1 append, filetype=0 output =1 i
nput
30090 ON ERROR GOTO 30150:IF FILEAPP=0 THEN 30110 ELSE FILEAPP=0
30100 OPEN FILENAM$ FOR APPEND AS #FILENUM:GOTO 30140
30110 IF FILETPE=1 THEN 30130
30120 OPEN FILENAM$ FOR OUTPUT AS #FILENUM:GOTO 30140
30130 OPEN FILENAM$ FOR INPUT AS #FILENUM
30140 FERR$="OK"
30145 ON ERROR GOTO 0:RETURN
30150 GOSUB 31260:RESUME 30145
30170 '
30180 'FILE DIRECTORY SUBROUTINE
30190 ON ERROR GOTO 30280
30200 COLOR 12,0:CLS:GOSUB 30030:LOCATE 1,18,0:IF SND%>1 THEN BEEP
30210 PRINT"WHICH DRIVE? (A, B, OR AB for both) >";:COLOR 6,0:INPU
T"",COM$
30220 PRINT:COLOR 3,0
30230 IF COM$="a:" OR COM$="A:" OR COM$="a" OR COM$="A" THEN PRINT"
A:\":FILES"A:*. *":GOTO 30310
30240 IF COM$="b:" OR COM$="B:" OR COM$="b" OR COM$="B" THEN PRINT"
B:\":FILES"B:*. *":GOTO 30310
30250 IF COM$="ab:" OR COM$="AB:" OR COM$="ab" OR COM$="AB" THEN 30
260 ELSE 30270
30260 PRINT"A:\":FILES"A:*. *":PRINT"B:\":FILES"B:*. *":GOTO 30310
30270 GOSUB 30760:GOTO 30200
30280 GOSUB 31260:GOSUB 30760:COLOR 15,0:PRINT:PRINT SPC(30) FERR$:
COLOR 7,0
30290 PRINT:PRINT SPC(31) RT$:INFLG=1:GOSUB 7120
30300 IF C$="Y" THEN RESUME 30200 ELSE RESUME 30310
30310 ON ERROR GOTO 0
30320 YFLD=24:GOSUB 31680:LOCATE 24,28,0:COLOR 12,0:PRINT"KILL ANY
FILES? (Y/N)";:INFLG=1:GOSUB 7120:IF C$="N" OR ESCK=1 THEN RETURN
30330 GOSUB 31680:LOCATE 24,32,0:PRINT"FILENAME >";:COLOR 6,0
30340 YFLD=24:XFLD=43:NUMDIG=14:INFLG=4:GOSUB 7110:IF ESCK=1 THEN R
ETURN
30350 FILENAM$=S$:GOSUB 31530:IF FERR$="OK" THEN 30320 ELSE GOSUB 3

```

```

1680:GOSUB 30760
30360 COLOR 15,0:LOCATE 24,13,0:PRINT FERR$;:LOCATE 24,47,0:COLOR 7
,0:PRINT RT$;
30370 INFLG=1:GOSUB 7120:IF C$="N" OR ESCK=1 THEN RETURN ELSE COLOR
12,0:GOTO 30330
30380 '
30390 'PROGRAM WRITE TO ARCHIVE
30400 'Calling routine provides COM$
30410 FILENAM$=ARCNAM$:FILENUM=2:FILEAPP=1:FILETPE=0
30420 GOSUB 30090:IF FERR$<>"OK" THEN CLOSE #FILENUM:GOTO 30450 'op
en file
30430 ON ERROR GOTO 30490:PRINT #FILENUM,COM$
30440 CLOSE #FILENUM:ON ERROR GOTO 0:RETURN
30450 CLS:COLOR 15,0:LOCATE 10,28,0:PRINT"ERROR ARCHIVING SESSION"
30460 LOCATE 12,28,0:PRINT FERR$:COLOR 7,0:GOSUB 30760
30470 LOCATE 14,30,0:PRINT RT$:INFLG=1:GOSUB 7120
30480 IF C$="Y" THEN 30420 ELSE RETURN
30490 ON ERROR GOTO 0:GOSUB 31260:RESUME 30450
30500 '
30510 'FILENAME INFORMATION & DIRECTORY ROUTINE
30520 COLOR 3,0:CLS
30530 LOCATE 2,15,0:PRINT"A filename is 1-8 letters/digits. A file
name may"
30540 LOCATE 3,15,0:PRINT"be preceded by a disk drive letter and a
colon,"
30550 LOCATE 4,15,0:PRINT"otherwise the current logged drive is use
d."
30560 IF FILETPE=0 THEN COLOR 15,0:LOCATE 6,8,0:PRINT"WARNING! Usi
ng an existing filename will erase the existing file."
30570 COLOR 3,0:LOCATE 8,24,0:PRINT"Would you like to see your"
30580 COLOR 12,0:LOCATE 9,26,0:PRINT"File Directory? (Y/N)"
30590 INFLG=1:GOSUB 7120:IF C$="Y" THEN GOSUB 30190
30600 RETURN
30610 '
30620 'INPUT FILENAME ROUTINE
30630 'calling routine supplies filext$, filenum, filetype, fileapp
30640 'leaves file open if OK, closes if FO=1
30650 CLS:COLOR 3,0:LOCATE 2,24,0:PRINT"Enter filename including dr
ive."
30660 LOCATE 3,20,0:PRINT"File will be given the extension "FILEXT$
30670 LOCATE 5,30,0:COLOR 12,0:PRINT"FILENAME >"
30680 XFLD=41:YFLD=5:NUMDIG=10:INFLG=4:GOSUB 7110:IF ESCK=1 THEN RE
TURN
30690 FILENAM$=S$+FILEXT$:GOSUB 30090
30700 IF FERR$="OK" THEN RETURN ELSE CLOSE #FILENUM
30710 COLOR 15,0:GOSUB 30760:LOCATE 7,28,0:PRINT FERR$:COLOR 7,0
30720 LOCATE 9,30,0:PRINT RT$:INFLG=1:GOSUB 7120
30730 IF C$="Y" THEN 30650 ELSE RETURN
30740 '
30750 'FLUB SOUND SUBROUTINE
30760 IF SND%<1 THEN RETURN ELSE SOUND 1000,2:SOUND 1180,3:BEEP:RET

```

```

URN
30770 '
30780 'PRESS ANY KEY ROUTINE
30790 'Caller specifies XFLD & YFLD coordinates of message
30800 COLOR 7,0: LOCATE YFLD,XFLD,0: PRINT"Press any key.....";:GOS
UB 30030
30810 IF SND%>1 THEN BEEP
30820 C$=INKEY$: IF C$="" THEN 30820
30830 IF C$=CHR$(27) THEN ESCK=1 ELSE ESCK=0
30840 RETURN
30850 COLOR 7,0: PRINT CHR$(13) CHR$(13) SPC(29)"Press any key.....
.";:GOSUB 30030:GOTO 30810
30860 '
30870 'INPUT/PRINT MATRIX ROUTINE
30880 'Caller specifies MAT, XFLD, YFLD, NUMDIG, P=1 PRINT
30890 COLOR 12,0: LOCATE YFLD,XFLD,0: PRINT COM$: YFLD=YFLD+2: XFLD=1: I
NFLG=3
30900 FOR I=1 TO ROW%: XOFF=6: NC=0: COLOR 12,0: LOCATE YFLD,XFLD: PRINT
"R";
30910 LOCATE YFLD,XFLD+1: PRINT I;: COLOR 6,0: XFLD=6: FOR J=1 TO COL%:
NC=NC+1
30920 IF P=1 THEN 30980 ELSE GOSUB 7110: IF ESCK=1 THEN RETURN
30930 IF MAT=1 THEN A(I,J)=S: GOTO 31040
30940 IF MAT=2 THEN B(I,J)=S: GOTO 31040
30950 IF MAT=3 THEN C(I,J)=S: GOTO 31040
30960 IF MAT=4 THEN D(I,J)=S: GOTO 31040
30970 IF MAT=5 THEN XO(I)=S: GOTO 31040
30980 LOCATE YFLD,XFLD,0
30990 IF MAT=1 THEN PRINT A(I,J);:GOTO 31040
31000 IF MAT=2 THEN PRINT B(I,J);:GOTO 31040
31010 IF MAT=3 THEN PRINT C(I,J);:GOTO 31040
31020 IF MAT=4 THEN PRINT D(I,J);:GOTO 31040
31030 IF MAT=5 THEN PRINT XO(I);:GOTO 31040
31035 IF MAT=6 THEN PRINT SPO(I,J);
31040 XFLD=NC*(NUMDIG+2)+XOFF: IF XFLD>70 THEN YFLD=YFLD+1: XFLD=10: X
OFF=10: NC=0: IF YFLD>22 THEN GOSUB 31070
31050 NEXT J: YFLD=YFLD+1: IF YFLD>22 THEN GOSUB 31070
31060 XFLD=1: NEXT I: RETURN
31070 SXFLD=XFLD: XFLD=28: YFLD=24: GOSUB 30800: YFLD=3: XFLD=SXFLD: COLO
R 6,0: CLS: RETURN
31080 '
31090 'READ MATRIX FROM DISK FILE
31100 'Caller specifies MAT, FILENUM, ROW%, COL%, & opens file
31110 ON ERROR GOTO 31220
31120 FOR I=1 TO ROW%: FOR J=1 TO COL%: IF EOF(FILENUM) THEN 31210
31130 IF MAT=0 THEN INPUT #FILENUM,KDIM(I): GOTO 31200
31140 IF MAT=1 THEN INPUT #FILENUM,A(I,J): GOTO 31200
31150 IF MAT=2 THEN INPUT #FILENUM,B(I,J): GOTO 31200
31160 IF MAT=3 THEN INPUT #FILENUM,C(I,J): GOTO 31200
31170 IF MAT=4 THEN INPUT #FILENUM,D(I,J): GOTO 31200
31180 IF MAT=5 THEN INPUT #FILENUM,XO(I): GOTO 31200

```

```

31190 IF MAT=6 THEN INPUT #FILENUM,OP(I,J)
31200 NEXT J,I:FERR$="OK":RETURN
31210 FERR$="END OF FILE BEFORE FINISHED":RETURN
31220 ON ERROR GOTO 0:GOSUB 31260:RESUME 31230
31230 RETURN
31240 '
31250 'ERROR CHECKING ROUTINE
31260 IF ERR=64 THEN FERR$="BAD FILE NAME":RETURN
31270 IF ERR=70 THEN FERR$="DISK WRITE PROTECT":RETURN
31280 IF ERR=53 THEN FERR$="FILE NOT FOUND":RETURN
31290 IF ERR=67 THEN FERR$="TOO MANY FILES ON DISK":RETURN
31300 IF ERR=55 THEN FERR$="FILE ALREADY IN USE":RETURN
31310 IF ERR=71 THEN FERR$="DISK NOT READY":RETURN
31320 IF ERR=61 THEN FERR$="DISK FULL":RETURN
31330 IF ERR=72 THEN FERR$="BAD DISK MEDIA":RETURN
31340 FERR$="DISK I/O ERROR":RETURN
31350 '
31360 'SAVE MATRIX ON DISK
31370 'Caller specifies MAT, FILENUM, ROW%, COL% & opens file
31380 ON ERROR GOTO 31480
31390 FOR I=1 TO ROW%:FOR J=1 TO COL%
31400 IF MAT=0 THEN PRINT #FILENUM,KDIM(I);:GOTO 31470
31410 IF MAT=1 THEN PRINT #FILENUM,A(I,J);:GOTO 31470
31420 IF MAT=2 THEN PRINT #FILENUM,B(I,J);:GOTO 31470
31430 IF MAT=3 THEN PRINT #FILENUM,C(I,J);:GOTO 31470
31440 IF MAT=4 THEN PRINT #FILENUM,D(I,J);:GOTO 31470
31450 IF MAT=5 THEN PRINT #FILENUM,X0(I);:GOTO 31470
31460 IF MAT=6 THEN PRINT #FILENUM,OP(I,J);
31470 NEXT J,I:FERR$="OK":ON ERROR GOTO 0:RETURN
31480 ON ERROR GOTO 0:GOSUB 31260:RESUME 31490
31490 RETURN
31500 '
31510 'KILL FILE ROUTINE
31520 'Caller specifies FILENAM$
31530 ON ERROR GOTO 31550:KILL FILENAM$:FERR$="OK"
31540 ON ERROR GOTO 0:RETURN
31550 GOSUB 31260:RESUME 31540
31570 '
31580 'CHANGE MATRIX ELEMENT
31590 'Caller specifies MAT, ROW1%, COL1%, & S (the new value)
31600 IF MAT=0 THEN KDIM(ROW1%)=S:RETURN
31610 IF MAT=1 THEN A(ROW1%,COL1%)=S:RETURN
31620 IF MAT=2 THEN B(ROW1%,COL1%)=S:RETURN
31630 IF MAT=3 THEN C(ROW1%,COL1%)=S:RETURN
31640 IF MAT=4 THEN D(ROW1%,COL1%)=S:RETURN
31650 IF MAT=5 THEN X0(ROW1%)=S:RETURN
31660 '
31670 'ERASE ANY LINE ROUTINE
31680 LOCATE YFLD,1,0:PRINT SPC(79);:RETURN
31690 '
31700 'WRITE BLANK TO ARCHIVE

```

```

31710 COM$="":GOSUB 30410:RETURN
31720 '
31730 'STATE TRANSITION MATRIX CALCULATION
31740 L%=50:E=.001:FOR I=1 TO KDIM(1):SX(I)=X0(I):FOR J=1 TO KDIM(1)
)
31750 IF I=J THEN SDI(I,J)=1 ELSE SDI(I,J)=0
31760 SP1(I,J)=T*SDI(I,J):SG(I,J)=T*A(I,J):NEXT J,I:N%=1
31770 N%=N%+1:FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1)
31780 SW(I,J)=T/N%*SG(I,J):SP0(I,J)=SP1(I,J):SP1(I,J)=SP1(I,J)+SW(I,J)
31790 SG(I,J)=0:NEXT J,I
31800 FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1):FOR K%=1 TO KDIM(1)
31810 SG(I,J)=SG(I,J)+SW(I,K%)*A(K%,J):NEXT K%,J,I
31820 IF N%<L% THEN 31860 ELSE GOSUB 30760:CLS:COLOR 15,0
31830 COM$="STATE TRANSITION MATRIX NOT CONVERGING PROPERLY!":IF AR
C%=1 THEN GOSUB 30410
31840 LOCATE 10,15,0:PRINT COM$:COLOR 7,0:LOCATE 12,13,0:PRINT"Plea
se re-evaluate model or try smaller time step"
31850 XFID=28:YFID=14:GOSUB 30800:RS=0:RETURN
31860 FOR Q%=1 TO KDIM(1):FOR R%=1 TO KDIM(1)
31870 IF SP1(Q%,R%)=0 THEN 31890
31880 IF E<ABS((SP1(Q%,R%)-SP0(Q%,R%))/SP1(Q%,R%)) THEN 31770
31890 NEXT R%,Q%
31900 FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1):SG(I,J)=0:FOR K%=1 TO K
DIM(1)
31910 SG(I,J)=SG(I,J)+A(I,K%)*SP1(K%,J):NEXT K%,J,I
31920 FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1):SP0(I,J)=SDI(I,J)+SG(I,J)
31930 NEXT J,I:RETURN
31933 '
32300 'NO MODEL/MATRIX ERROR ROUTINE
32310 GOSUB 30760:CLS:COLOR 15,0:LOCATE 10,28,0:PRINT COM$:XFID=28:
YFID=12:GOSUB 30800:RETURN
32320 '
32330 'ERASE EXISTING ERROR ROUTINE
32340 GOSUB 30760:CLS:COLOR 15,0:LOCATE 10,24,0:PRINT COM$:COLOR 7,
0:INFLG=1:GOSUB 7120:RETURN
32350 '
32430 'MODIFY ANY MATRIX ROUTINE
32440 CLS:XFID=1:YFID=3:NUMDIG=10:P=1:GOSUB 30890:COLOR 12,0
32450 LOCATE 25,1,0:PRINT"<ENTER> re-input matrix, ESC quit, else R
OW >";
32460 LOCATE 25,46,0:PRINT SPC(33):XFID=48:YFID=25:NUMDIG=10:INFLG=
3
32470 GOSUB 7110:IF ESCK=1 THEN RETURN
32480 IF S<>0 THEN 32510 ELSE XFID=1:YFID=3:CLS:P=0:GOSUB 30890
32490 IF ESCK<1 THEN RETURN ELSE CLS:COLOR 15,0:LOCATE 10,27,0
32500 PRINT"WARNING! MATRIX INCOMPLETE":XFID=30:YFID=12:GOSUB 3076
0:GOSUB 30800:RETURN
32510 IF S>ROW% OR S<1 THEN GOSUB 30760:GOTO 32460
32520 ROW1%=S:LOCATE 25,46,0:PRINT ROW1% SPC(5);:LOCATE 25,50,0:COL

```

```
OR 12,0:PRINT"COL >";
32530 LOCATE 25,55,0:PRINT SPC(23):XFLD=56:YFLD=25:GOSUB 7110:IF ES
CK=1 THEN RETURN
32540 IF S>COL% OR S<1 THEN GOSUB 30760:GOTO 32530
32550 COL1%=S:LOCATE 25,55,0:PRINT COL1% SPC(6);:LOCATE 25,59,0:COL
OR 12,0:PRINT"VALUE >";:XFLD=67:YFLD=25
32560 GOSUB 7110:IF ESCK=1 THEN RETURN ELSE GOSUB 31600:GOTO 32440
32590 '
32700 GOSUB 30760:CLS:COLOR 15,0:LOCATE 10,28,0:PRINT COM$:COLOR 7,
0
32710 LOCATE 12,33,0:PRINT RT$:INFLG=1:GOSUB 7120:RETURN
```

C.5 RAT 3/6/85

```

10 COMMON HELP%,HELP$,ARC%,SND%,QW,ARCNAM$: OPTION BASE 1: DEFINT I,J
30 RAT$="-----TIME RESPONSE-----"
40 DIM A(10,10),B(10),C(10),D(20,20),X0(20),SG(20,20),SW(20,20)
50 DIM SPO(20,20),SDI(21),KDIM(6),NS(10),DS(11),AA(20,20),CAA1(20,2
0)
55 DIM SDJ(21),K(10),COEF(21),CF(20,20,20),CAA2(20,20),CAA3(20,20)
60 IF QW<1 THEN HELP%=0: HELP$="OFF": ARC%=0: SND%=1
72 MAXDIM=10: DIGIT$="1234567890": RT$="RETRY? (Y/N)"
74 DEF FNCRM(R1,AI1,R2,AI2)=R1*R2-AI1*AI2
75 DEF FNCIM(R1,AI1,R2,AI2)=R1*AI2+R2*AI1
76 DEF FNCRD(R1,AI1,R2,AI2)=(R1*R2+AI1*AI2)/(R2+AI2)
77 DEF FNCID(R1,AI1,R2,AI2)=(AI1*R2-R1*AI2)/(R2+AI2)
80 EM=0: IF ARC%=1 THEN GOSUB 31710: COM$=RAT$: GOSUB 30410: GOSUB 3171
0
90 CLS: WIDTH 80: COLOR 14,0: LOCATE 3,30,0: PRINT "TIME RESPONSE"
100 COLOR 10,0: LOCATE 6,23,0: PRINT "<F1>.....ENTER NEW MODEL"
110 LOCATE 7,23,0: PRINT "<F2>.....RECALL MODEL/INPUT"
120 LOCATE 8,23,0: PRINT "<F3>.....MODIFY MODEL/INPUT"
130 LOCATE 9,23,0: PRINT "<F4>.....STORE MODEL/INPUT"
140 LOCATE 10,23,0: PRINT "<F5>.....ENTER INPUT PARAMETERS"
150 LOCATE 11,23,0: PRINT "<F6>.....CALCULATE DATA"
170 LOCATE 12,23,0: PRINT "<F8>.....LIST/SAVE/RECALL DATA"
180 LOCATE 13,23,0: PRINT "<F9>.....FILE DIRECTORY"
190 LOCATE 14,23,0: PRINT "<ESC>.....EXIT TO MAIN MENU"
200 LOCATE 15,23,0: PRINT "<Alt-F1>...HELP LEVEL"
210 LOCATE 16,23,0: PRINT "<Alt-F2>...SOUND LEVEL"
220 COLOR 14,0: LOCATE 15,46,0: PRINT HELP$: LOCATE 16,46,0: PRINT SND%
230 COLOR 10,0: LOCATE 17,23,0: PRINT "<Alt-F3>...ARCHIVE/COMMENTS"
240 COLOR 14,0: LOCATE 20,24,0: PRINT "PRESS DESIRED FUNCTION KEY"
250 INFLG=2: GOSUB 7120: IF ESCK=1 AND QW=1 THEN CHAIN "DACS"
255 IF ESCK=1 AND QW<>1 THEN 383      'system
260 IF C=59 THEN GOSUB 850: GOTO 90    'new model
270 IF C=60 THEN GOSUB 1540: GOTO 90   'recall model
280 IF C=61 THEN GOSUB 4260: GOTO 90   'modify
290 IF C=62 THEN GOSUB 1300: GOTO 90   'store model
300 IF C=63 THEN GOSUB 1928: GOTO 90   'input params
310 IF C=64 THEN GOSUB 2500: GOTO 90   'run
330 IF C=66 THEN GOSUB 3590: GOTO 90   'list
340 IF C=67 THEN GOSUB 30190: GOTO 90  'files
350 IF C=104 THEN GOSUB 590: GOTO 220  'help
360 IF C=105 THEN GOSUB 550: GOTO 220  'help
370 IF C=106 THEN GOSUB 390: GOTO 90   'archive
380 GOSUB 30760: GOTO 250
381 '
382 ' SYSTEM
383 CLEAR: WIDTH 80: SCREEN 0,0: COLOR 7,0: CLOSE: END
385 '
390 ' ARCHIVE
392 IF ARC%=1 THEN GOSUB 420: RETURN

```

```

394 GOSUB 420: IF ARC%=0 THEN RETURN ELSE GOSUB 31710: COM$=RAT$: GOSU
B 30410: GOSUB 31710: RETURN
396 '
410 'ARCHIVE SESSION / ADD COMMENTS
420 COLOR 12,0:CLS: IF ARC%=1 THEN 470 ELSE LOCATE 2,27,0
430 PRINT"ARCHIVE SESSION? (Y/N)": INFLG=1: GOSUB 7120: IF ESCK=1 OR
C$="N" THEN RETURN
440 FILETPE=0: IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
450 FILEXT$=".ARC": FILENUM=2: GOSUB 30650: CLOSE #FILENUM
460 IF ESCK=1 OR FERR$<>"OK" THEN RETURN ELSE ARCNAM$=FILENAM$: ARC%
=1: CLS
470 COLOR 12,0: LOCATE 2,28,0: PRINT"ENTER COMMENTS? (Y/N)": INFLG=1:
GOSUB 7120
480 IF C$="N" OR ESCK=1 THEN RETURN ELSE NUMDIG=78: INFLG=5: XFLD=1
490 CLS: LOCATE 2,19,0: YFLD=4
500 PRINT"ENTER COMMENTS BELOW....PRESS ESC TO QUIT": PRINT
510 GOSUB 7110: IF ESCK=1 THEN GOSUB 31710: RETURN
520 COM$=S$: GOSUB 30410: YFLD=YFLD+1: XFLD=1: IF YFLD>24 THEN 490 ELSE
510
530 '
540 'SOUND LEVEL
550 SND%=SND%+1: IF SND%>2 THEN SND%=0
560 RETURN
570 '
580 'SET HELP LEVEL
590 IF HELP%=1 THEN HELP%=0: HELP$="OFF": RETURN
600 HELP%=1: HELP$="ON ": RETURN
610 '
840 'ENTER NEW MODEL ROUTINE
850 IF EM=1 THEN COM$="ERASE EXISTING MODEL? (Y/N)": GOSUB 32340: IF
C$="N" OR ESCK=1 THEN RETURN
880 CLS: IF HELP%=0 THEN 990
882 COLOR 3,0: LOCATE 1,12,0: PRINT"This program calculates the time
response in closed form"
884 LOCATE 2,12,0: PRINT"for the closed-loop system"
886 COLOR 15,0: LOCATE 4,27,0: PRINT"x(t) = Ax(t) + bu(t)"
888 LOCATE 5,30,0: PRINT"u(t) = KU[r(t) - kx(t)]"
890 LOCATE 6,30,0: PRINT"y(t) = cx(t)": COLOR 3,0
892 LOCATE 8,12,0: PRINT"due to any initial conditions x(0) and inpu
t r(t), t>=0"
894 LOCATE 9,12,0: PRINT"which has a rational Laplace transform R(s)
with a pole-"
896 LOCATE 10,12,0: PRINT"zero excess of at least one. You will be
prompted for"
898 LOCATE 11,12,0: PRINT"the system dimension (DA) and the constant
matrices, A, b,"
900 LOCATE 12,12,0: PRINT"c, k, and x(0). x(0) is the initial condi
tion vector and"
902 LOCATE 13,12,0: PRINT"the matrices have dimensions"
904 COLOR 15,0: LOCATE 15,36,0: PRINT"A(DA,DA)": LOCATE 16,36,0: PRINT"
b(DA,1)"

```

```

906 LOCATE 17,36,0:PRINT"c(1,DA)":LOCATE 18,36,0:PRINT"k(1,DA)":COL
OR 3,0
908 LOCATE 20,12,0:PRINT"The input transfer function R(s) and the i
nput gain KU"
910 LOCATE 21,12,0:PRINT"are entered under INPUT PARAMETERS."
912 GOSUB 30850:IF ESCK=1 THEN 1250
990 CLS:GOSUB 1030:IF ESCK=1 THEN 1250
1010 GOSUB 1160:IF ESCK=1 THEN 1250
1020 EM=1:IF ARC%=0 THEN RETURN ELSE GOSUB 31710:COM$="-ENTER NEW M
ODEL-":GOSUB 30410:RETURN
1030 COLOR 12,0:LOCATE 10,29,0:PRINT"SYSTEM DIMENSION >"SPC(38)
1040 XFLD=49:YFLD=10:NUMDIG=12:INFLG=3:GOSUB 7110:IF ESCK=1 THEN RE
TURN
1050 S=INT(S):IF S<=MAXDIM AND S>0 THEN KDIM(1)=S:RETURN
1060 GOSUB 30760:COLOR 15,0:LOCATE 12,18,0:PRINT"ERROR!  Value must
lie in the range 1 to"MAXDIM:GOTO 1030
1160 CLS:NUMDIG=10:INFLG=3:P=0:IF HELP%=0 THEN 1180 ELSE COLOR 3,0
1170 LOCATE 1,6,0:PRINT"ENTER MATRICES BELOW BY ROWS PRESSING <ENTE
R> AFTER EACH ELEMENT."
1180 MAT=1:COM$="A MATRIX":XFLD=1:YFLD=3:ROW%=KDIM(1):COL%=KDIM(1):
GOSUB 30890:IF ESCK=1 THEN RETURN
1200 CLS:MAT=2:COM$="b MATRIX":XFLD=1:YFLD=3:ROW%=KDIM(1):COL%=1:GO
SUB 30890:IF ESCK=1 THEN RETURN
1210 CLS:MAT=3:COM$="c MATRIX":XFLD=1:YFLD=3:ROW%=1:COL%=KDIM(1):GO
SUB 30890:IF ESCK=1 THEN RETURN
1230 CLS:MAT=4:COM$="k MATRIX":XFLD=1:YFLD=3:ROW%=1:COL%=KDIM(1):GO
SUB 30890:IF ESCK=1 THEN RETURN
1240 CLS:MAT=5:COM$="X0 MATRIX":XFLD=1:YFLD=3:ROW%=KDIM(1):COL%=1:G
OSUB 30890:RETURN
1250 COM$="MODEL NOT ENTERED!!":GOSUB 32310:EM=0:RETURN
1270 '
1280 'SAVE MODEL ROUTINE
1300 IF IP=0 AND EM=0 THEN COM$="NOTHING TO SAVE!!":GOSUB 32310:RET
URN
1304 CLS:COLOR 10,0:LOCATE 9,27,0:PRINT"<Alt-F5>....SAVE MODEL"
1305 LOCATE 10,27,0:PRINT"<Alt-F6>....SAVE INPUTS"
1306 LOCATE 11,27,0:PRINT"<ESC>.....END"
1307 INFLG=2:GOSUB 7120:IF ESCK=1 THEN RETURN
1308 IF C=108 THEN GOSUB 1329:GOTO 1304
1309 IF C=109 THEN GOSUB 1512:GOTO 1304
1310 GOSUB 30760:GOTO 1307
1329 IF EM=0 THEN COM$="NO MODEL TO SAVE!!":GOSUB 32310:RETURN
1330 FILETPE=0:IF HELP%=1 THEN GOSUB 30520 ELSE CLS:GOSUB 30580
1340 FILEXT$=".RTM":FILENUM=1:GOSUB 30650
1350 IF ESCK=0 AND FERR$="OK" THEN 1380
1360 COM$="MODEL NOT SAVED!!":GOSUB 32310:RETURN
1380 SFILE$=FILENAM$:CLS:COLOR 3,0:LOCATE 10,30,0:PRINT"SAVING MODE
L....."
1390 IF ARC%=1 THEN GOSUB 31710:COM$="-STORE MODEL ON DISK-":GOSUB
30410:FILENUM=1
1400 MAT=0:ROW%=1:COL%=1:GOSUB 31380:IF FERR$<>"OK" THEN 1480

```

```

1410 MAT=1: ROW%=KDIM(1): COL%=KDIM(1): GOSUB 31380: IF FERR$<>"OK" THE
N 1480
1420 MAT=2: ROW%=KDIM(1): COL%=1: GOSUB 31380: IF FERR$<>"OK" THEN 1480
1430 MAT=3: ROW%=1: COL%=KDIM(1): GOSUB 31380: IF FERR$<>"OK" THEN 1480
1440 MAT=4: ROW%=1: COL%=KDIM(1): GOSUB 31380: IF FERR$<>"OK" THEN 1480
1450 MAT=5: ROW%=KDIM(1): COL%=1: GOSUB 31380: IF FERR$<>"OK" THEN 1480
1460 CLOSE #FILENAME: IF ARC%=1 THEN COM$="Model saved under filename
"+SFILE$: GOSUB 30410
1470 RETURN
1480 CLS: LOCATE 10,28,0: COLOR 15,0: PRINT FERR$: COLOR 7,0: GOSUB 3076
0
1490 FILENAM$=SFILE$: GOSUB 31530: LOCATE 12,33,0: PRINT RT$: INFLG=1: G
OSUB 7120
1500 IF C$="Y" AND ESCK=0 THEN 1330
1510 IF ARC%=0 THEN 1360 ELSE COM$=FERR$: GOSUB 30410: GOTO 1360
1511 '
1512 IF IP=0 THEN COM$="NO INPUTS TO SAVE!!": GOSUB 32310: RETURN
1513 FILETPE=0: IF HELP%=1 THEN GOSUB 30520 ELSE CLS: GOSUB 30580
1514 FILEXT$=".RTI": FILENUM=1: GOSUB 30650
1515 IF ESCK=0 AND FERR$="OK" THEN 1517
1516 COM$="INPUTS NOT SAVED!!": GOSUB 32310: RETURN
1517 SFILE$=FILENAM$: CLS: COLOR 3,0: LOCATE 10,30,0: PRINT "SAVING INPU
TS.....": IF ARC%=1 THEN COM$="-STORE INPUTS ON DISK-": GOSUB 30410:
FILENUM=1
1518 MAT=0: ROW%=5: COL%=1: GOSUB 31380: IF FERR$<>"OK" THEN 1523
1519 MAT=7: ROW%=1: COL%=KDIM(4)+1: GOSUB 31380: IF FERR$<>"OK" THEN 15
23
1520 MAT=8: ROW%=1: COL%=KDIM(5)+1: GOSUB 31380: IF FERR$<>"OK" THEN 15
23
1521 CLOSE #FILENAME: IF ARC%=1 THEN COM$="Inputs saved under filenam
e "+SFILE$: GOSUB 30410
1522 RETURN
1523 CLS: LOCATE 10,28,0: COLOR 15,0: PRINT FERR$: COLOR 7,0: GOSUB 3076
0
1524 FILENAM$=SFILE$: GOSUB 31530: LOCATE 12,33,0: PRINT RT$: INFLG=1: G
OSUB 7120
1525 IF C$="Y" AND ESCK=0 THEN 1513
1526 IF ARC%=0 THEN 1516 ELSE COM$=FERR$: GOSUB 30410: GOTO 1516
1527 '
1530 'RECALL MODEL ROUTINE
1540 CLS: COLOR 10,0: LOCATE 9,27,0: PRINT "<Alt-F7>....RECALL MODEL"
1542 LOCATE 10,27,0: PRINT "<Alt-F8>....RECALL INPUTS"
1544 LOCATE 11,27,0: PRINT "<ESC>.....END"
1546 INFLG=2: GOSUB 7120: IF ESCK=1 THEN RETURN
1548 IF C=110 THEN GOSUB 1565: GOTO 1540
1550 IF C=111 THEN GOSUB 1882: GOTO 1540
1552 GOSUB 30760: GOTO 1546
1565 IF EM=1 THEN COM$="ERASE EXISTING MODEL? (Y/N)": GOSUB 32340: I
F C$="N" OR ESCK=1 THEN RETURN
1570 FILETPE=1: IF HELP%=1 THEN GOSUB 30520 ELSE CLS: GOSUB 30580
1580 FILEXT$=".RTM": FILENUM=1: GOSUB 30650

```

```

1590 IF ESCK=0 AND FERR$="OK" THEN SFILE$=FILENAM$:GOTO 1620
1600 COM$="MODEL NOT RECALLED!!":GOSUB 32310:RETURN
1620 CLS:COLOR 3,0:LOCATE 10,30,0:PRINT"RECALLING MODEL....."
1630 IF ARC%=1 THEN GOSUB 31710:COM$="-RECALL MODEL-":GOSUB 30410:F
ILENUM=1
1640 ROW%=1:COL%=1:MAT=0:GOSUB 31110:IF FERR$<>"OK" THEN 1840
1650 ROW%=KDIM(1):COL%=KDIM(1):MAT=1:GOSUB 31110:IF FERR$<>"OK" THE
N 1840
1660 ROW%=KDIM(1):COL%=1:MAT=2:GOSUB 31110:IF FERR$<>"OK" THEN 1840
1670 ROW%=1:COL%=KDIM(1):MAT=3:GOSUB 31110:IF FERR$<>"OK" THEN 1840
1680 ROW%=1:COL%=KDIM(1):MAT=4:GOSUB 31110:IF FERR$<>"OK" THEN 1840
1690 ROW%=KDIM(1):COL%=1:MAT=5:GOSUB 31110:IF FERR$<>"OK" THEN 1840
ELSE EM=1
1700 CLOSE #FILENUM:COLOR 7,0:LOCATE 14,30,0:PRINT"SEE MODEL? (Y/N
)"
1710 INFLG=1:GOSUB 7120:IF C$="Y" AND ESCK=0 THEN 1740
1720 IF ARC%=1 THEN COM$="Model recalled from file "+SFILE$:GOSUB 3
0410
1730 RETURN
1740 CLS:COLOR 12,0:LOCATE 10,30,0:PRINT"SYSTEM DIMENSION >":COLOR
6,0:LOCATE 10,49,0:PRINT KDIM(1)
1760 GOSUB 30850:IF ESCK=1 THEN 1720 ELSE P=1:NUMDIG=10:CLS
1770 YFLD=3:XFLD=1:MAT=1:COM$="A MATRIX":ROW%=KDIM(1):COL%=KDIM(1):
GOSUB 30890:GOSUB 30850:IF ESCK=1 THEN 1720
1790 CLS:YFLD=3:XFLD=1:MAT=2:COM$="b MATRIX":ROW%=KDIM(1):COL%=1:GO
SUB 30890:GOSUB 30850:IF ESCK=1 THEN 1720
1800 CLS:YFLD=3:XFLD=1:MAT=3:COM$="c MATRIX":ROW%=1:COL%=KDIM(1):GO
SUB 30890:GOSUB 30850:IF ESCK=1 THEN 1720
1820 CLS:YFLD=3:XFLD=1:MAT=4:COM$="k MATRIX":ROW%=1:COL%=KDIM(1):GO
SUB 30890:GOSUB 30850:IF ESCK=1 THEN 1720
1830 CLS:YFLD=3:XFLD=1:MAT=5:COM$="XO MATRIX":ROW%=KDIM(1):COL%=1:G
OSUB 30890:GOSUB 30850:GOTO 1720
1840 CLOSE #FILENUM:CLS:COLOR 15,0:GOSUB 30760:LOCATE 10,28,0:PRINT
FERR$
1850 LOCATE 12,33,0:COLOR 7,0:PRINT RT$:INFLG=1:GOSUB 7120
1860 IF C$="Y" AND ESCK=0 THEN 1570 ELSE EM=0
1870 IF ARC%=1 THEN COM$=FERR$+" for file "+SFILE$:GOSUB 30410
1880 GOTO 1600
1881 '
1882 IF IP=1 THEN COM$="ERASE EXISTING INPUTS? (Y/N)":GOSUB 32340:
IF C$="N" OR ESCK=1 THEN RETURN
1883 FILETPE=1:IF HELP%=1 THEN GOSUB 30520 ELSE CLS:GOSUB 30580
1884 FILEXT$=".RTI":FILENUM=1:GOSUB 30650
1885 IF ESCK=0 AND FERR$="OK" THEN SFILE$=FILENAM$:GOTO 1887
1886 COM$="INPUTS NOT RECALLED!!":GOSUB 32310:RETURN
1887 CLS:COLOR 3,0:LOCATE 10,30,0:PRINT"RECALLING INPUTS....."
1888 IF ARC%=1 THEN GOSUB 31710:COM$="-RECALL INPUTS-":GOSUB 30410:
FILENUM=1
1889 ROW%=5:COL%=1:MAT=0:GOSUB 31110:IF FERR$<>"OK" THEN 1903
1890 ROW%=1:COL%=KDIM(4)+1:MAT=7:GOSUB 31110:IF FERR$<>"OK" THEN 19
03

```

```

1891 ROW%=1:COL%=KDIM(5)+1:MAT=8:GOSUB 3110:IF FERR$<>"OK" THEN 19
03 ELSE IP=1
1892 CLOSE #FILENUM:COLOR 7,0:LOCATE 14,30,0:PRINT"SEE INPUTS? (Y/
N)"
1893 INFLG=1:GOSUB 7120:IF C$="Y" AND ESCK=0 THEN 1896
1894 IF ARC%=1 THEN COM$="Inputs recalled from file "+SFILE$:GOSUB
30410
1895 RETURN
1896 CLS:COLOR 12,0:LOCATE 6,34,0:PRINT"GAIN KU >":COLOR 6,0:LOCATE
6,44,0:PRINT KDIM(2)
1897 COLOR 12,0:LOCATE 8,34,0:PRINT"GAIN KR >":COLOR 6,0:LOCATE 8,4
4,0:PRINT KDIM(3)
1898 COLOR 12,0:LOCATE 10,27,0:PRINT"DIMENSION N(s) >":COLOR 6,0:LO
CATE 10,44,0:PRINT KDIM(4)
1899 COLOR 12,0:LOCATE 12,27,0:PRINT"DIMENSION D(s) >":COLOR 6,0:LO
CATE 12,44,0:PRINT KDIM(5)
1900 GOSUB 30850:IF ESCK=1 THEN 1894 ELSE P=1:NUMDIG=10:CLS
1901 YFLD=3:XFLD=1:MAT=7:COM$="N(s) POLYNOMIAL":ROW%=1:COL%=KDIM(4)
+1:GOSUB 30890:GOSUB 30850:IF ESCK=1 THEN 1894
1902 CLS:YFLD=3:XFLD=1:MAT=8:COM$="D(s) POLYNOMIAL":ROW%=1:COL%=KDI
M(5)+1:GOSUB 30890:GOSUB 30850:GOTO 1894
1903 CLOSE #FILENUM:CLS:COLOR 15,0:GOSUB 30760:LOCATE 10,28,0:PRINT
FERR$
1904 LOCATE 12,33,0:COLOR 7,0:PRINT RT$:INFLG=1:GOSUB 7120
1905 IF C$="Y" AND ESCK=0 THEN 1883 ELSE IP=0
1906 IF ARC%=1 THEN COM$=FERR$+" for file "+SFILE$:GOSUB 30410
1907 GOTO 1886
1908 '
1927 ' INPUT PARAMETERS
1928 IF EM=0 THEN COM$="ENTER MODEL FIRST!!":GOSUB 32310:RETURN
1929 CLS:IP=0:NUMDIG=12:IF HELP%=0 THEN 2080 ELSE COLOR 3,0
1930 LOCATE 1,12,0:PRINT"The input to the system is a combination o
f an external"
1940 LOCATE 2,12,0:PRINT"input and a feedback of system states, suc
h that"
1950 COLOR 15,0:LOCATE 4,30,0:PRINT"u(t) = KU[r(t) - kx(t)"] :COLOR
3,0
1960 LOCATE 6,12,0:PRINT"r(t) is the external input and is describe
d by a"
1970 LOCATE 7,12,0:PRINT"rational Laplace transform of the form"
1980 COLOR 15,0:LOCATE 9,30,0:PRINT"R(s) = KR*N(s)/D(s)":COLOR 3,0
1990 LOCATE 11,12,0:PRINT"You will be prompted below for the gains
KU and KR, the"
2000 LOCATE 12,12,0:PRINT"dimensions of N(s) and D(s) and the polyn
omials N(s)
2010 LOCATE 13,12,0:PRINT"and D(s). The coefficient of the highest
order term of"
2020 LOCATE 14,12,0:PRINT"the polynomials is assumed unity and is n
ot entered."
2030 LOCATE 15,12,0:PRINT"Enter the coefficients in ascending order
. Example:"

```

```

2040 COLOR 15,0: LOCATE 17,20,0: PRINT "R(s) = (s+3)/(s**3 +2s**2 + 9s
+ 3)"
2050 LOCATE 19,30,0: PRINT "N(s) = 3": LOCATE 20,30,0: PRINT "D(s) = 3
9 2"
2070 GOSUB 30850: IF ESCK=1 THEN 2260 ELSE CLS
2080 COLOR 12,0: LOCATE 3,34,0: PRINT "GAIN KU >" SPC(12)
2090 XFLD=44: YFLD=3: NUMDIG=12: INFLG=3: GOSUB 7110: IF ESCK=1 THEN 226
0 ELSE KDIM(2)=S
2100 COLOR 12,0: LOCATE 7,34,0: PRINT "GAIN KR >" SPC(12)
2110 XFLD=44: YFLD=7: NUMDIG=12: INFLG=3: GOSUB 7110: IF ESCK=1 THEN 226
0 ELSE KDIM(3)=S
2120 COLOR 12,0: LOCATE 11,27,0: PRINT "DIMENSION N(s) >" SPC(12)
2130 XFLD=44: YFLD=11: NUMDIG=12: INFLG=3: GOSUB 7110: IF ESCK=1 THEN 22
60
2140 S=INT(S): IF S>=0 AND S<=MAXDIM-1 THEN KDIM(4)=S: GOTO 2160
2150 GOSUB 30760: COLOR 15,0: LOCATE 13,18,0: PRINT "ERROR! Value must
lie in the range 0 to" MAXDIM-1: GOTO 2120
2160 COLOR 12,0: LOCATE 15,27,0: PRINT "DIMENSION D(S) >" SPC(12)
2170 XFLD=44: YFLD=15: NUMDIG=12: INFLG=3: GOSUB 7110: IF ESCK=1 THEN 22
60
2180 S=INT(S): IF S>=0 AND S<=MAXDIM AND S>KDIM(4) THEN KDIM(5)=S: GO
TO 2200
2190 GOSUB 30760: COLOR 15,0: LOCATE 17,18,0: PRINT "ERROR! Value must
lie in the range 0 to" MAXDIM
2195 LOCATE 18,18,0: PRINT "and be greater than the Dimension of N(s)
." : GOTO 2160
2200 CLS: NUMDIG=10: INFLG=3: P=0: IF HELP%=0 THEN 2220 ELSE COLOR 3,0
2210 LOCATE 1,3,0: PRINT "ENTER POLYNOMIALS IN ASCENDING ORDER PRESSI
NG <ENTER> AFTER EACH ELEMENT.";
2220 NS(KDIM(4)+1)=1: DS(KDIM(5)+1)=1: IF KDIM(4)=0 THEN 2240
2230 MAT=7: COM$="N(s) POLYNOMIAL": XFLD=1: YFLD=3: ROW%=1: COL%=KDIM(4)
: GOSUB 30890: IF ESCK=1 THEN 2260
2240 MAT=8: COM$="D(s) POLYNOMIAL": XFLD=1: YFLD=8: ROW%=1: COL%=KDIM(5)
: GOSUB 30890: IF ESCK=1 THEN 2260 ELSE IP=1
2250 IF ARC%<>1 THEN RETURN ELSE GOSUB 31710: COM$="-ENTER INPUT PAR
AMETERS-": GOSUB 30410: RETURN
2260 COM$="INPUTS NOT ENTERED!!": GOSUB 32310: IP=0: RETURN
2480 '
2490 'RUN SIMULATION CALCULATE DATA
2500 IF EM=0 THEN COM$="ENTER MODEL FIRST!!": GOSUB 32310: RETURN
2510 IF IP=0 THEN COM$="ENTER INPUTS FIRST!!": GOSUB 32310: RETURN
2540 RS=1: CLS: LOCATE 11,29,0: COLOR 3,0: PRINT "CALCULATING DATA.....
"
2550 IF ARC%=1 THEN GOSUB 31710: COM$="-RUN SIMULATION-": GOSUB 30410
2560 TD=KDIM(1)+KDIM(5): FOR I=1 TO TD: FOR J=1 TO TD: AA(I,J)=0: NEXT
J,I
2570 FOR I=1 TO KDIM(1): FOR J=1 TO KDIM(1): AA(I,J)=A(I,J)-KDIM(2)*B
(I)*K(J): NEXT J,I
2580 FOR I=1 TO KDIM(1): FOR J=KDIM(1)+1 TO KDIM(1)+KDIM(4)+1: AA(I,J
)=KDIM(2)*B(I)*KDIM(3)*NS(J-KDIM(1)): NEXT J,I
2590 FOR I=KDIM(1)+1 TO KDIM(1)+KDIM(5): FOR J=KDIM(1)+1 TO KDIM(1)+

```

```

KDIM(5): IF J=I+1 THEN AA(I,J)=1
2600 NEXT J,I: FOR I=1 TO KDIM(5): AA(KDIM(1)+KDIM(5),KDIM(1)+I)=-DS(
I): NEXT I
2610 GOSUB 32930: IF RS>0 THEN 2640 ELSE COLOR 15,0: GOSUB 30760: LOC
ATE 13,18,0
2620 PRINT"ERROR CALCULATING CHARACTERISTIC POLYNOMIAL": GOSUB 30850
: RETURN
2640 N=TD: FOR I=1 TO N+1: SDI(I)=COEF(I): NEXT I: GOSUB 33170: N=TD
2650 GOSUB 34010: IF RS>0 THEN 2660 ELSE ROW%=TD: COL%=2: P=1: MAT=6: YF
LD=3
2652 XFLD=1: CLS: COM$="THE EIGENVALUES ARE": GOSUB 30890: GOSUB 30850:
RETURN
2660 FOR I=KDIM(1)+1 TO TD: X0(I)=0: NEXT I: X0(TD)=1
2670 FOR I=1 TO TD: FOR J=1 TO KDIM(1): SG(J,I)=0: FOR K=1 TO TD
2680 SG(J,I)=SG(J,I)+CF(I,J,K)*X0(K): NEXT K,J,I
2690 FOR I=1 TO TD: SDI(I)=0: FOR J=1 TO KDIM(1)
2700 SDI(I)=SDI(I)+SG(J,I)*C(J): NEXT J,I: RETURN
3570 '
3580 ' SAVE/LIST DATA
3590 IF ARC%=1 THEN GOSUB 31710: COM$="-SAVE/LIST DATA-": GOSUB 30410
3600 CLS: COLOR 14,0: LOCATE 4,36,0: PRINT"OPTIONS": COLOR 10,0
3610 LOCATE 6,25,0: PRINT"<Ctrl-F6>....LIST CURRENT DATA"
3620 LOCATE 7,25,0: PRINT"<Ctrl-F7>....SAVE DATA ON DISK"
3630 LOCATE 8,25,0: PRINT"<Ctrl-F8>....RECALL DATA FILE"
3640 LOCATE 9,25,0: PRINT"<ESC>.....END"
3650 INFLG=2: GOSUB 7120: IF ESCK=1 THEN RETURN
3660 IF C=99 THEN GOSUB 3720: GOTO 3600
3670 IF C=100 THEN GOSUB 3890: GOTO 3600
3680 IF C=101 THEN GOSUB 4070: GOTO 3600
3690 GOSUB 30760: GOTO 3650
3700 '
3710 ' LIST CURRENT DATA
3720 IF RS=0 THEN COM$="NO DATA TO LIST!!": GOSUB 32310: RETURN
3740 CLS: P=1: IF ARC%=1 THEN COM$="Listed current data": GOSUB 30410
3750 MAT=11: ROW%=KDIM(1): COL%=1: T$="STATE RESPONSE: VECTOR COEFFIC
IENT OF EXP("
3760 FOR IP=1 TO TD: YFLD=1: XFLD=1: FOR J=1 TO KDIM(1): SDJ(J)=SG(J,IP
): NEXT J
3770 IF ABS(SPO(IP,2))>.00001 THEN 3790 ELSE COM$=T$+STR$(SPO(IP,1)
)+")t IS": CLS
3780 GOSUB 30890: GOSUB 30850: IF ESCK=1 THEN RETURN ELSE 3830
3790 IS=IP+1: COM$=T$+STR$(SPO(IS,1))+")t*cos("+STR$(SPO(IS,2))+")t
IS": CLS: GOSUB 30890
3800 GOSUB 30850: IF ESCK=1 THEN RETURN ELSE FOR J=1 TO KDIM(1): SDJ(
J)=SG(J,IS): NEXT J
3810 COM$=T$+STR$(SPO(IS,1))+")T*SIN("+STR$(SPO(IS,2))+")t IS"
3820 CLS: YFLD=1: XFLD=1: GOSUB 30890: GOSUB 30850: IF ESCK=1 THEN RETUR
N ELSE IP=IP+1
3830 NEXT IP: CLS: COLOR 12,0: LOCATE 1,1,0: PRINT"RESPONSE OF THE OUTP
UT Y(t) IS THE SUMMATION"
3840 COLOR 6,0: FOR I=1 TO TD: LOCATE 2+I,1,0

```

```

3845 IF ABS(SPO(I,2))<.00001 THEN PRINT SDI(I)*EXP("SPO(I,1)")t":G
OTO 3865
3850 PRINT SDI(I)*EXP("SPO(I+1,1)")t*cos("SPO(I+1,2)")t":I=I+1
3860 PRINT SDI(I)*EXP("SPO(I,1)")t*sin("SPO(I,2)")t"
3865 NEXT I:GOSUB 30850:RETURN
3870 '
3880 'SAVE DATA ON DISK
3890 IF RS=0 THEN COM$="NO DATA TO SAVE!!":GOSUB 32310:RETURN
3910 CLS:FILETPE=0:IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
3920 FILEXT$=".RAD":FILENUM=1:GOSUB 30650
3930 IF ESK=0 AND FERR$="OK" THEN 3960
3940 COM$="DATA NOT SAVED!!":GOSUB 32310:RETURN
3960 SFILE$=FILENAM$:CLS:COLOR 3,0:LOCATE 11,31,0:PRINT"SAVING DATA
....."
3970 MAT=0:ROW%=5:COL%=1:GOSUB 31380:IF FERR$<>"OK" THEN 4010
3975 MAT=6:ROW%=TD:COL%=2:GOSUB 31380:IF FERR$<>"OK" THEN 4010
3980 MAT=9:ROW%=KDIM(1):COL%=TD:GOSUB 31380:IF FERR$<>"OK" THEN 401
0
3985 MAT=10:ROW%=1:COL%=TD:GOSUB 31380:IF FERR$<>"OK" THEN 4010 ELS
E CLOSE #FILENUM
3990 IF ARC%=1 THEN COM$="Data saved in file "+SFILE$:GOSUB 30410
4000 RETURN
4010 CLOSE #FILENUM:CLS:GOSUB 30760:COLOR 15,0:LOCATE 10,28,0:PRINT
FERR$
4020 COLOR 7,0:LOCATE 12,33:PRINT RT$:INFLG=1:GOSUB 7120:IF C$="Y"
AND ESK=0 THEN 3910
4030 IF ARC%=1 THEN COM$=FERR$+" while saving data":GOSUB 30410
4040 FILENAM$=SFILE$:GOSUB 31530:RESUME 3940
4050 '
4060 'RECALL DATA FILE
4070 IF RS=1 THEN COM$="ERASE EXISTING DATA? (Y/N)":GOSUB 32340:IF
C$="N" OR ESK=1 THEN RETURN
4100 CLS:FILETPE=1:IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
4110 FILEXT$=".RAD":FILENUM=1:GOSUB 30650
4120 IF ESK=0 AND FERR$="OK" THEN SFILE$=FILENAM$:GOTO 4160
4130 COM$="DATA NOT RECALLED!!":GOSUB 32310:RETURN
4160 CLS:COLOR 3,0:LOCATE 10,29,0:PRINT"RECALLING DATA....."
4170 ROW%=5:COL%=1:MAT=0:GOSUB 31110:IF FERR$<>"OK" THEN 4210
4175 TD=KDIM(1)+KDIM(5):ROW%=TD:COL%=2:MAT=6:GOSUB 31110:IF FERR$<>
"OK" THEN 4210
4180 ROW%=KDIM(1):COL%=TD:MAT=9:GOSUB 31110:IF FERR$<>"OK" THEN 421
0
4185 ROW%=1:COL%=TD:MAT=10:GOSUB 31110:IF FERR$<>"OK" THEN 4210
4190 RS=1:CLOSE #FILENUM:IF ARC%=1 THEN COM$="Recalled data from fi
le "+SFILE$:GOSUB 30410
4200 RETURN
4210 CLOSE #FILENUM:GOSUB 30760:CLS:COLOR 15,0:LOCATE 10,28,0:PRINT
FERR$
4220 COLOR 7,0:LOCATE 12,33,0:PRINT RT$:INFLG=1:GOSUB 7120
4230 IF C$="Y" AND ESK=0 THEN 4100 ELSE RS=0:GOTO 4130
4240 '

```

```

4250 'MODIFY CURRENT MODEL
4260 IF EM=0 AND IP=0 THEN COM$='NOTHING TO MODIFY!!':GOSUB 32310:R
RETURN
4280 IF ARC%=1 THEN GOSUB 31710:COM$='-MODIFY MODEL/INPUT-':GOSUB 3
0410
4290 CLS: IF HELP%=0 THEN 4310 ELSE LOCATE 1,1,0:COLOR 3,0
4300 PRINT'ENTER LETTER OF MATRIX, S FOR SYSTEM DIMENSIONS, G FOR I
NPUT GAINS, ESC QUIT.'
4310 COLOR 12,0:LOCATE 3,19,0:PRINT'(A, b, c, k, X, S, N, D, G, OR
ESC) >'SPC(3):INFLG=5:NUMDIG=1:YFLD=3:XFLD=57:GOSUB 7110
4320 IF ESCK=1 THEN RETURN
4325 IF EM=0 THEN 4382
4330 IF S$='S' OR S$='s' THEN GOSUB 4400:GOTO 4290
4340 IF S$='A' OR S$='a' THEN COM$='A MATRIX':MAT=1:ROW%=KDIM(1):CO
L%=KDIM(1):GOSUB 32440:GOTO 4290
4350 IF S$='B' OR S$='b' THEN COM$='b MATRIX':MAT=2:ROW%=KDIM(1):CO
L%=1:GOSUB 32440:GOTO 4290
4360 IF S$='C' OR S$='c' THEN COM$='c MATRIX':MAT=3:ROW%=1:COL%=KDI
M(1):GOSUB 32440:GOTO 4290
4370 IF S$='K' OR S$='k' THEN COM$='k MATRIX':MAT=4:ROW%=1:COL%=KDI
M(1):GOSUB 32440:GOTO 4290
4380 IF S$='X' OR S$='x' THEN COM$='X0 MATRIX':MAT=5:ROW%=KDIM(1):C
OL%=1:GOSUB 32440:GOTO 4290
4382 IF IP=0 THEN 4390
4384 IF S$='N' OR S$='n' THEN COM$='N(s) POLYNOMIAL':MAT=7:ROW%=1:C
OL%=KDIM(4):IF COL%=0 THEN 4390 ELSE GOSUB 32440:GOTO 4290
4386 IF S$='D' OR S$='d' THEN COM$='D(s) POLYNOMIAL':MAT=8:ROW%=1:C
OL%=KDIM(5):IF COL%=0 THEN 4390 ELSE GOSUB 32440:GOTO 4290
4388 IF S$='G' OR S$='g' THEN GOSUB 4440:GOTO 4290
4390 GOSUB 30760:GOTO 4310
4400 CLS:LOCATE 1,1,0:COLOR 15,0:PRINT'Changing dimensions without
changing matrices can lead to unexpected results!!'
4430 GOSUB 1030:RETURN
4440 CLS:COLOR 12,0:LOCATE 10,34,0:PRINT'GAIN KU >':XFLD=44:YFLD=10
:NUMDIG=12
4442 INFLG=3:GOSUB 7110:IF ESCK=1 THEN RETURN ELSE KDIM(2)=S
4444 COLOR 12,0:LOCATE 14,34,0:PRINT'GAIN KR >':XFLD=44:YFLD=14:NUM
DIG=12
4446 INFLG=3:GOSUB 7110:IF ESCK=1 THEN RETURN ELSE KDIM(3)=S:RETURN
7030 '
7040 'INPUT ROUTINE for numbers, names, function keys, comments, &
Y/N answers
7110 COLOR 6,0
7120 SXFLD=XFLD:S$='':C$='':T$='':N$='':ESCK=0:GOSUB 30030:IF SND%>
1 THEN BEEP
7130 IF INFLG>2 THEN 7150
7140 C$=INKEY$:IF C$=' ' THEN 7140 ELSE 7170
7150 LOCATE YFLD,XFLD,1
7160 C$=INKEY$:IF C$=' ' THEN 7150
7170 IF C$=CHR$(27) THEN ESCK=1:RETURN
7180 IF INFLG=2 THEN 7370

```

'escape key

```

7190 IF LEN(C$)>1 THEN 7450 'invalid
7200 C=ASC(C$)
7210 IF INFLG=1 THEN 7320
7220 IF C=8 THEN 7500
7230 IF INFLG=5 THEN 7800 'comment
7240 IF INFLG=3 THEN 7390 'number
7250 'Name Routine
7260 IF C=13 THEN RETURN
7270 IF C=46 AND NUMDIG>10 THEN 7470 'okay
7280 IF C=91 OR C=92 OR C=93 OR C=124 THEN 7450 'invalid
7290 IF C=34 OR C=39 OR C=42 OR C=43 OR C=46 OR C=47 THEN 7450 'in
valid
7300 IF (C>32 AND C<59) OR (C>63 AND C<127) THEN 7470 ELSE 7450
7310 'Yes/No routine
7320 IF C=89 OR C=121 THEN C$="Y":RETURN
7330 IF C=78 OR C=110 THEN C$="N":RETURN
7340 IF C=13 THEN C$="N":RETURN
7350 GOTO 7450
7360 'Function Key
7370 IF LEN(C$)<2 THEN 7450 ELSE C$=RIGHT$(C$,1):C=ASC(C$)
7380 IF (C>58 AND C<69) OR (C>83 AND C<114) THEN RETURN ELSE 7450
7390 'Number Routine
7400 IF C=13 THEN 7540 'carriage return
7410 IF C>47 AND C<58 THEN 7470 'okay
7420 IF C=46 OR C=43 OR C=45 OR C=32 THEN 7470 'okay
7430 IF C=69 OR C=101 THEN C$=CHR$(69):GOTO 7470 'okay
7440 'Illegal character
7450 GOSUB 30760:BEEP:GOTO 7130
7460 'Character okay
7470 IF LEN(S$)>=NUMDIG THEN 7450
7480 LOCATE YFLD,XFLD:PRINT C$;:S$=S$+C$:XFLD=XFLD+1:GOTO 7130
7490 'Backspace
7500 XFLD=XFLD-1:LOCATE YFLD,XFLD:PRINT " ";:IF XFLD<=SXFLD THEN XFLD=SXFLD
7510 LNT=LEN(S$)-1:IF LNT<0 THEN LNT=0
7520 S$=MID$(S$,1,LNT):GOTO 7130
7530 'Number Check routine
7540 NUMCK=0:DECPT=0:EXPNT=0:SIGN=0
7550 IF S$="" THEN 7770 ELSE T$=LEFT$(S$,1)
7560 IF T$="+" OR T$="-" THEN N$=T$:S$=MID$(S$,2)
7570 IF T$="." THEN N$=T$:DECPT=1:S$=MID$(S$,2)
7580 IF S$="" THEN 7790
7590 IF (T$="+" OR T$="-") AND INSTR(".",LEFT$(S$,1))=1 THEN DECPT=1:T$=LEFT$(S$,1):N$=N$+T$:S$=MID$(S$,2)
7600 IF INSTR(DIGIT$,LEFT$(S$,1))=0 THEN 7790
7610 WHILE S$<>"":T$=LEFT$(S$,1):S$=MID$(S$,2)
7620 IF T$="" THEN 7680
7630 IF (T$="+" OR T$="-") AND RIGHT$(N$,1)<>"E" THEN NUMCK=1:GOTO 7680
7640 IF T$="." THEN IF DECPT=1 OR EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:DECPT=1

```

```

7650 IF T$="+" OR T$="-" THEN IF SIGN=1 OR EXPNT=0 THEN NUMCK=1 ELSE
E N$=N$+T$:SIGN=1
7660 IF T$="E" THEN IF EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:EXPNT=1:D
ECPT=1
7670 IF T$<>"." AND T$<> "+" AND T$<> "-" AND T$<> "E" THEN N$=N$+T$
7680 WEND: IF NUMCK=1 THEN 7790
7690 'Check for overflow
7700 D=0:N%=INSTR(N$,"E"): IF N%=0 THEN 7770
7710 IF N%=1 THEN 7790
7720 S$=LEFT$(N$,N%-1):C$=MID$(N$,N%+1):S=VAL(S$)
7730 IF S=0 THEN D=0 ELSE D=INT(LOG(ABS(S))/LOG(10))
7740 IF C$<>"" THEN D=D+VAL(C$)
7750 IF D<-37 OR D>37 THEN 7790
7760 'convert
7770 S=VAL(N$):RETURN
7780 'invalid
7790 NUMCK=1:GOSUB 30760:BEEP:LOCATE YFLD,SXFLD,0:PRINT SPC(NUMDIG)
:XFLD=SXFLD:GOTO 7120
7800 'Comment routine
7810 IF C$=CHR$(13) THEN RETURN
7820 IF LEN(S$)>=NUMDIG THEN RETURN ELSE 7480
7830 '
30000 '***** SUBROUTINES *****
30010 '
30020 'CLEAR KBD BUFFER
30030 DEF SEG=0:POKE 1050,PEEK(1052):DEF SEG
30040 RETURN
30050 '
30060 'OPEN DISK FILE FOR INPUT/OUTPUT
30090 ON ERROR GOTO 30150:IF FILEAPP=0 THEN 30110 ELSE FILEAPP=0
30100 OPEN FILENAM$ FOR APPEND AS #FILENUM:GOTO 30140
30110 IF FILETPE=1 THEN 30130
30120 OPEN FILENAM$ FOR OUTPUT AS #FILENUM:GOTO 30140
30130 OPEN FILENAM$ FOR INPUT AS #FILENUM
30140 FERR$="OK"
30145 ON ERROR GOTO 0:RETURN
30150 GOSUB 31260:RESUME 30145
30170 '
30180 'FILE DIRECTORY SUBROUTINE
30190 ON ERROR GOTO 30280
30200 COLOR 12,0:CLS:GOSUB 30030:LOCATE 1,18,0:IF SND%>1 THEN BEEP
30210 PRINT"WHICH DRIVE? (A, B, OR AB for both) >";:COLOR 6,0:INPU
T"",COM$
30220 PRINT:COLOR 3,0
30230 IF COM$="a:" OR COM$="A:" OR COM$="a" OR COM$="A" THEN PRINT"
A:\":FILES"A:*. *":GOTO 30310
30240 IF COM$="b:" OR COM$="B:" OR COM$="b" OR COM$="B" THEN PRINT"
B:\":FILES"B:*. *":GOTO 30310
30250 IF COM$="ab:" OR COM$="AB:" OR COM$="ab" OR COM$="AB" THEN 30
260 ELSE 30270
30260 PRINT"A:\":FILES"A:*. *":PRINT"B:\":FILES"B:*. *":GOTO 30310

```

```

30270 GOSUB 30760:GOTO 30200
30280 GOSUB 31260:GOSUB 30760:COLOR 15,0:PRINT:PRINT SPC(30) FERR$:
COLOR 7,0
30290 PRINT:PRINT SPC(31) RT$:INFLG=1:GOSUB 7120
30300 IF C$="Y" THEN RESUME 30200 ELSE RESUME 30310
30310 ON ERROR GOTO 0
30320 YFLD=24:GOSUB 31680:LOCATE 24,28,0:COLOR 12,0:PRINT"KILL ANY
FILES? (Y/N)";:INFLG=1:GOSUB 7120:IF C$="N" OR ESCK=1 THEN RETURN
30330 GOSUB 31680:LOCATE 24,32,0:PRINT"FILENAME >";:COLOR 6,0
30340 YFLD=24:XFLD=43:NUMDIG=14:INFLG=4:GOSUB 7110:IF ESCK=1 THEN R
ETURN
30350 FILENAM$=S$:GOSUB 31530:IF FERR$="OK" THEN 30320 ELSE GOSUB 3
1680:GOSUB 30760
30360 COLOR 15,0:LOCATE 24,13,0:PRINT FERR$;:LOCATE 24,47,0:COLOR 7
,0:PRINT RT$;
30370 INFLG=1:GOSUB 7120:IF C$="N" OR ESCK=1 THEN RETURN ELSE COLOR
12,0:GOTO 30330
30380 '
30390 'PROGRAM WRITE TO ARCHIVE
30410 FILENAM$=ARCNAM$:FILENUM=2:FILEAPP=1:FILETPE=0
30420 GOSUB 30090:IF FERR$<>"OK" THEN CLOSE #FILENUM:GOTO 30450 'op
en file
30430 ON ERROR GOTO 30490:PRINT #FILENUM,COM$
30440 CLOSE #FILENUM:ON ERROR GOTO 0:RETURN
30450 CLS:COLOR 15,0:LOCATE 10,28,0:PRINT"ERROR ARCHIVING SESSION"
30460 LOCATE 12,28,0:PRINT FERR$:COLOR 7,0:GOSUB 30760
30470 LOCATE 14,30,0:PRINT RT$:INFLG=1:GOSUB 7120
30480 IF C$="Y" THEN 30420 ELSE RETURN
30490 ON ERROR GOTO 0:GOSUB 31260:RESUME 30450
30500 '
30510 'FILENAME INFORMATION & DIRECTORY ROUTINE
30520 COLOR 3,0:CLS
30530 LOCATE 2,15,0:PRINT"A filename is 1-8 letters/digits. A file
name may"
30540 LOCATE 3,15,0:PRINT"be preceded by a disk drive letter and a
colon,"
30550 LOCATE 4,15,0:PRINT"otherwise the current logged drive is use
d."
30560 IF FILETPE=0 THEN COLOR 15,0:LOCATE 6,8,0:PRINT"WARNING! Usi
ng an existing filename will erase the existing file."
30570 COLOR 3,0:LOCATE 8,24,0:PRINT"Would you like to see your"
30580 COLOR 12,0:LOCATE 9,26,0:PRINT"File Directory? (Y/N)"
30590 INFLG=1:GOSUB 7120:IF C$="Y" THEN GOSUB 30190
30600 RETURN
30610 '
30620 'INPUT FILENAME ROUTINE
30650 CLS:COLOR 3,0:LOCATE 2,24,0:PRINT"Enter filename including dr
ive."
30660 LOCATE 3,20,0:PRINT"File will be given the extension "FILEXT$
30670 LOCATE 5,30,0:COLOR 12,0:PRINT"FILENAME >"
30680 XFLD=41:YFLD=5:NUMDIG=10:INFLG=4:GOSUB 7110:IF ESCK=1 THEN RE

```

TURN

```

30690 FILENAM$=S$+FILEXT$:GOSUB 30090
30700 IF FERR$="OK" THEN RETURN ELSE CLOSE #FILENUM
30710 COLOR 15,0:GOSUB 30760:LOCATE 7,28,0:PRINT FERR$:COLOR 7,0
30720 LOCATE 9,30,0:PRINT RT$:INFLG=1:GOSUB 7120
30730 IF C$="Y" THEN 30650 ELSE RETURN
30740 '
30750 'FLUB SOUND SUBROUTINE
30760 IF SND%<1 THEN RETURN ELSE SOUND 1000,2:SOUND 1180,3:BEEP:RET
URN
30770 '
30780 'PRESS ANY KEY ROUTINE
30800 COLOR 7,0:LOCATE YFLD,XFLD,0:PRINT"Press any key.....";:GOS
UB 30030
30810 IF SND%>1 THEN BEEP
30820 C$=INKEY$:IF C$="" THEN 30820
30830 IF C$=CHR$(27) THEN ESCK=1 ELSE ESCK=0
30840 RETURN
30850 COLOR 7,0:PRINT CHR$(13) CHR$(13) SPC(29)"Press any key.....
.";:GOSUB 30030:GOTO 30810
30860 '
30870 'INPUT/PRINT MATRIX ROUTINE
30890 COLOR 12,0:LOCATE YFLD,XFLD,0:PRINT COM$:YFLD=YFLD+2:XFLD=1:I
NFLG=3
30900 FOR I=1 TO ROW%:XOFF=6:NC=0:COLOR 12,0:LOCATE YFLD,XFLD:PRINT
"R";
30910 LOCATE YFLD,XFLD+1:PRINT I;:COLOR 6,0:XFLD=6:FOR J=1 TO COL%:
NC=NC+1
30920 IF P=1 THEN 30980 ELSE GOSUB 7110:IF ESCK=1 THEN RETURN
30930 IF MAT=1 THEN A(I,J)=S:GOTO 31040
30940 IF MAT=2 THEN B(I)=S:GOTO 31040
30950 IF MAT=3 THEN C(J)=S:GOTO 31040
30960 IF MAT=4 THEN K(J)=S:GOTO 31040
30970 IF MAT=5 THEN X0(I)=S:GOTO 31040
30972 IF MAT=7 THEN NS(J)=S:GOTO 31040
30974 IF MAT=8 THEN DS(J)=S:GOTO 31040
30980 LOCATE YFLD,XFLD,0
30990 IF MAT=1 THEN PRINT A(I,J);:GOTO 31040
31000 IF MAT=2 THEN PRINT B(I);:GOTO 31040
31010 IF MAT=3 THEN PRINT C(J);:GOTO 31040
31020 IF MAT=4 THEN PRINT K(J);:GOTO 31040
31030 IF MAT=5 THEN PRINT X0(I);:GOTO 31040
31035 IF MAT=6 THEN PRINT SPO(I,J);:GOTO 31040
31036 IF MAT=7 THEN PRINT NS(J);:GOTO 31040
31037 IF MAT=8 THEN PRINT DS(J);:GOTO 31040
31038 IF MAT=11 THEN PRINT SDJ(I);
31040 XFLD=NC*(NUMDIG+2)+XOFF:IF XFLD>70 THEN YFLD=YFLD+1:XFLD=10:X
OFF=10:NC=0:IF YFLD>22 THEN GOSUB 31070
31050 NEXT J:YFLD=YFLD+1:IF YFLD>22 THEN GOSUB 31070
31060 XFLD=1:NEXT I:RETURN
31070 SXFLD=XFLD:XFLD=28:YFLD=24:GOSUB 30800:YFLD=3:XFLD=SXFLD:COLO

```

```

R 6,0:CLS:RETURN
31080 '
31090 'READ MATRIX FROM DISK FILE
31110 ON ERROR GOTO 31220
31120 FOR I=1 TO ROW%:FOR J=1 TO COL%:IF EOF(FILENUM) THEN 31210
31130 IF MAT=0 THEN INPUT #FILENUM,KDIM(I):GOTO 31200
31140 IF MAT=1 THEN INPUT #FILENUM,A(I,J):GOTO 31200
31150 IF MAT=2 THEN INPUT #FILENUM,B(I):GOTO 31200
31160 IF MAT=3 THEN INPUT #FILENUM,C(J):GOTO 31200
31170 IF MAT=4 THEN INPUT #FILENUM,K(J):GOTO 31200
31180 IF MAT=5 THEN INPUT #FILENUM,X0(I):GOTO 31200
31185 IF MAT=6 THEN INPUT #FILENUM,SP0(I,J):GOTO 31200
31192 IF MAT=7 THEN INPUT #FILENUM,NS(J):GOTO 31200
31194 IF MAT=8 THEN INPUT #FILENUM,DS(J):GOTO 31200
31196 IF MAT=9 THEN INPUT #FILENUM,SG(I,J):GOTO 31200
31197 IF MAT=10 THEN INPUT #FILENUM,SDI(J)
31200 NEXT J,I:FERR$="OK":RETURN
31210 FERR$="END OF FILE BEFORE FINISHED":RETURN
31220 ON ERROR GOTO 0:GOSUB 31260:RESUME 31230
31230 RETURN
31240 '
31250 'ERROR CHECKING ROUTINE
31260 IF ERR=64 THEN FERR$="BAD FILE NAME":RETURN
31270 IF ERR=70 THEN FERR$="DISK WRITE PROTECT":RETURN
31280 IF ERR=53 THEN FERR$="FILE NOT FOUND":RETURN
31290 IF ERR=67 THEN FERR$="TOO MANY FILES ON DISK":RETURN
31300 IF ERR=55 THEN FERR$="FILE ALREADY IN USE":RETURN
31310 IF ERR=71 THEN FERR$="DISK NOT READY":RETURN
31320 IF ERR=61 THEN FERR$="DISK FULL":RETURN
31330 IF ERR=72 THEN FERR$="BAD DISK MEDIA":RETURN
31340 FERR$="DISK I/O ERROR":RETURN
31350 '
31360 'SAVE MATRIX ON DISK
31380 ON ERROR GOTO 31480
31390 FOR I=1 TO ROW%:FOR J=1 TO COL%
31400 IF MAT=0 THEN PRINT #FILENUM,KDIM(I);:GOTO 31470
31410 IF MAT=1 THEN PRINT #FILENUM,A(I,J);:GOTO 31470
31420 IF MAT=2 THEN PRINT #FILENUM,B(I);:GOTO 31470
31430 IF MAT=3 THEN PRINT #FILENUM,C(J);:GOTO 31470
31440 IF MAT=4 THEN PRINT #FILENUM,K(J);:GOTO 31470
31450 IF MAT=5 THEN PRINT #FILENUM,X0(I);:GOTO 31470
31460 IF MAT=6 THEN PRINT #FILENUM,SP0(I,J);:GOTO 31470
31462 IF MAT=7 THEN PRINT #FILENUM,NS(J);:GOTO 31470
31464 IF MAT=8 THEN PRINT #FILENUM,DS(J);:GOTO 31470
31465 IF MAT=9 THEN PRINT #FILENUM,SG(I,J);:GOTO 31470
31466 IF MAT=10 THEN PRINT #FILENUM,SDI(J);
31470 NEXT J,I:FERR$="OK":ON ERROR GOTO 0:RETURN
31480 ON ERROR GOTO 0:GOSUB 31260:RESUME 31490
31490 RETURN
31500 '
31510 'KILL FILE ROUTINE

```

```

31530 ON ERROR GOTO 31550: KILL FILENAM$
31540 ON ERROR GOTO 0: FERR$="OK": RETURN
31550 GOSUB 31260: RESUME 31560
31560 RETURN
31570 '
31580 'CHANGE MATRIX ELEMENT
31600 IF MAT=0 THEN KDIM(ROW1%)=S: RETURN
31610 IF MAT=1 THEN A(ROW1%,COL1%)=S: RETURN
31620 IF MAT=2 THEN B(ROW1%)=S: RETURN
31630 IF MAT=3 THEN C(COL1%)=S: RETURN
31640 IF MAT=4 THEN K(COL1%)=S: RETURN
31650 IF MAT=5 THEN XO(ROW1%)=S: RETURN
31652 IF MAT=7 THEN NS(COL1%)=S: RETURN
31654 IF MAT=8 THEN DS(COL1%)=S: RETURN
31660 '
31670 'ERASE ANY LINE ROUTINE
31680 LOCATE YFLD,1,0: PRINT SPC(79);: RETURN
31690 '
31700 'WRITE BLANK TO ARCHIVE
31710 COM$="": GOSUB 30410: RETURN
31720 '
32300 'NO MODEL/MATRIX ERROR ROUTINE
32310 GOSUB 30760: CLS: COLOR 15,0: LOCATE 10,28,0: PRINT COM$: XFLD=28:
YFLD=12: GOSUB 30800: RETURN
32320 '
32330 'ERASE EXISTING ERROR ROUTINE
32340 GOSUB 30760: CLS: COLOR 15,0: LOCATE 10,24,0: PRINT COM$: COLOR 7,
0: INFLG=1: GOSUB 7120: RETURN
32350 '
32430 'MODIFY ANY MATRIX ROUTINE
32440 CLS: XFLD=1: YFLD=3: NUMDIG=10: P=1: GOSUB 30890: COLOR 12,0
32450 LOCATE 25,1,0: PRINT"<ENTER> re-input matrix, ESC quit, else R
OW >";
32460 LOCATE 25,46,0: PRINT SPC(33): XFLD=48: YFLD=25: NUMDIG=10: INFLG=
3
32470 GOSUB 7110: IF ESCK=1 THEN RETURN
32480 IF S<>0 THEN 32510 ELSE XFLD=1: YFLD=3: CLS: P=0: GOSUB 30890
32490 IF ESCK<1 THEN RETURN ELSE CLS: COLOR 15,0: LOCATE 10,27,0
32500 PRINT"WARNING!  MATRIX INCOMPLETE": XFLD=30: YFLD=12: GOSUB 3076
0: GOSUB 30800: RETURN
32510 IF S>ROW% OR S<1 THEN GOSUB 30760: GOTO 32460
32520 ROW1%=S: LOCATE 25,46,0: PRINT ROW1% SPC(5);: LOCATE 25,50,0: COL
OR 12,0: PRINT"COL >";
32530 LOCATE 25,55,0: PRINT SPC(23): XFLD=56: YFLD=25: GOSUB 7110: IF ES
CK=1 THEN RETURN
32540 IF S>COL% OR S<1 THEN GOSUB 30760: GOTO 32530
32550 COL1%=S: LOCATE 25,55,0: PRINT COL1% SPC(6);: LOCATE 25,59,0: COL
OR 12,0: PRINT"VALUE >";: XFLD=67: YFLD=25
32560 GOSUB 7110: IF ESCK=1 THEN RETURN ELSE GOSUB 31600: GOTO 32440
32590 '
32700 GOSUB 30760: CLS: COLOR 15,0: LOCATE 10,28,0: PRINT COM$: COLOR 7,

```

0

```

32710 LOCATE 12,33,0:PRINT RT$:INFLG=1:GOSUB 7120:RETURN
32720 '
32730 'CALCULATE DETERMINANT
32780 IREV=0:FOR I=1 TO NN%:KKK%=I
32790 IF SG(KKK%,I)<>0 THEN 32810 ELSE KKK%=KKK%+1
32800 IF KKK%<=NN% THEN 32790 ELSE 32900
32810 IF I<KKK% THEN 32830
32820 IF I>KKK% THEN 32900 ELSE 32840
32830 FOR JJ=1 TO NN%:TEMP=SG(I,JJ):SG(I,JJ)=SG(KKK%,JJ):SG(KKK%,JJ)
)=TEMP:NEXT JJ:IREV=IREV+1
32840 II=I+1:IF II>NN% THEN 32880
32850 FOR JJ=II TO NN%:IF SG(JJ,I)=0 THEN 32870 ELSE TEMP=SG(JJ,I)/
SG(I,I)
32860 FOR N%=I TO NN%:SG(JJ,N%)=SG(JJ,N%)-SG(I,N%)*TEMP:NEXT N%
32870 NEXT JJ,I
32880 DET=1:FOR I=1 TO NN%:DET=DET*SG(I,I):NEXT I
32890 DET=(-1)CIREV*DET:RETURN
32900 DET=0:RETURN
32910 '
32920 'CALCULATE CHARACTERISTIC POLYNOMIAL
32930 FOR I=1 TO TD:COEF(I)=0:NEXT I:COEF(TD+1)=1:FOR J=1 TO TD:K%=
0:L%=1:D(1,1)=0
33030 D(L%,1)=D(L%,1)+1:IF (L%-J)<0 THEN 33050
33040 IF (L%-J)>0 THEN 33120 ELSE 33060
33050 FOR I=L% TO J-1:D(I+1,1)=D(I,1)+1:NEXT I
33060 FOR I=1 TO J:FOR KK%=1 TO J:ROW%=D(I,1):COL%=D(KK%,1):SG(I,KK
%)=AA(ROW%,COL%):NEXT KK%,I
33070 K%=K%+1:NN%=J:GOSUB 32780:SW(K%,1)=DET
33080 FOR I=1 TO J:L%=J-I+1:IF (D(L%,1)-(TD-J+L%))<0 THEN 33030
33090 IF (D(L%,1)-(TD-J+L%))>0 THEN 33120
33100 NEXT I:S%=TD-J+1
33110 FOR I=1 TO K%:COEF(S%)=COEF(S%)+SW(I,1)*(-1)CJ:NEXT I,J:RETUR
N
33120 RS=0:RETURN
33145 '
33150 'ROOT FINDING ROUTINE
33170 IF N<5 THEN 35000
33175 IREV=1:NC=N+1:FOR I=1 TO NC:SDJ(I)=SDI(I):NEXT I:P=0:Q=0:R=0
33180 IF SDJ(1)<>0 THEN 33200 ELSE NC=NC-1
33190 SPO(NC,1)=0:SPO(NC,2)=0:FOR I=1 TO NC:SDJ(I)=SDJ(I+1):NEXT I:
GOTO 33180
33200 IF (NC-1)=0 THEN RETURN
33210 IF (NC-2)=0 THEN R=-SDJ(1)/SDJ(2):GOTO 33430
33220 IF (NC-3)=0 THEN P=SDJ(2)/SDJ(3):Q=SDJ(1)/SDJ(3):GOTO 33450
33230 IF (ABS(SDJ(NC-1)/SDJ(NC))-ABS(SDJ(2)/SDJ(1)))>=0 THEN 33280
33240 IREV=-IREV:FOR I=1 TO NC/2:NL=NC+1-I:TEMP=SDJ(NL):SDJ(NL)=SDJ
(I):SDJ(I)=TEMP:NEXT I
33250 IF Q=0 THEN P=0:GOTO 33270
33260 P=P/Q:Q=1/Q
33270 IF R<>0 THEN R=1/R

```

```

33280 E=5E-10: SG(NC,1)=SDJ(NC): SDI(NC)=SDJ(NC): SG(NC+1,1)=0: SDI(NC+
1)=0: NP=NC-1
33290 FOR J=1 TO 1000: FOR II=1 TO NP: I=NC-II: SG(I,1)=SDJ(I)+R*SG(I+
1,1)
33300 SDI(I)=SG(I,1)+R*SDI(I+1): NEXT II
33310 IF (ABS(SG(1,1)/SDJ(1))-E)<=0 THEN 33430
33320 IF SDI(2)=0 THEN R=R+1 ELSE R=R-SG(1,1)/SDI(2)
33330 FOR II=1 TO NP: I=NC-II
33340 SG(I,1)=SDJ(I)-P*SG(I+1,1)-Q*SG(I+2,1): SDI(I)=SG(I,1)-P*SDI(I
+1)-Q*SDI(I+2): NEXT II
33350 IF SDJ(2)<>0 THEN 33370
33360 IF ABS(SG(2,1)/SDJ(1))-E<=0 THEN 33380 ELSE 33390
33370 IF ABS(SG(2,1)/SDJ(2))-E>0 THEN 33390
33380 IF ABS(SG(1,1)/SDJ(1))-E<=0 THEN 33450
33390 CBAR=SDI(2)-SG(2,1): D=SDI(3)*2-CBAR*SDI(4)
33400 IF D=0 THEN P=P-2: Q=Q*(Q+1): GOTO 33420
33410 P=P+(SG(2,1)*SDI(3)-SG(1,1)*SDI(4))/D: Q=Q+(-SG(2,1)*CBAR+SG(1
,1)*SDI(3))/D
33420 NEXT J: E=E*10: GOTO 33290
33430 NC=NC-1: SPO(NC,2)=0: IF IREV<0 THEN SPO(NC,1)=1/R ELSE SPO(NC,
1)=R
33440 FOR I=1 TO NC: SDJ(I)=SG(I+1,1): NEXT I: GOTO 33200
33450 NC=NC-2: IF IREV<0 THEN QP=1/Q: PP=P/(Q*2): GOTO 33470
33460 QP=Q: PP=P/2
33470 TEMP=PP*2-QP: IF TEMP>=0 THEN 33490
33480 SPO(NC+1,1)=-PP: SPO(NC,1)=-PP: SPO(NC+1,2)=SQR(-TEMP): SPO(NC,2
)=-SPO(NC+1,2): GOTO 33510
33490 IF PP=0 THEN SPO(NC+1,1)=-SQR(TEMP) ELSE SPO(NC+1,1)=(-PP/ABS
(PP))*(ABS(PP)+SQR(TEMP))
33500 SPO(NC+1,2)=0: SPO(NC,1)=QP/SPO(NC+1,1): SPO(NC,2)=0
33510 FOR I=1 TO NC: SDJ(I)=SG(I+2,1): NEXT I: GOTO 33200
33999 '
34000 ' SYLVESTER EXPANSION THEOREM
34010 FOR I=1 TO N: FOR J=I+1 TO N: IF ABS(SPO(I,1)-SPO(J,1))>.00001
OR ABS(SPO(I,2)-SPO(J,2))>.00001 THEN 34040
34020 COLOR 15,0: GOSUB 30760: LOCATE 13,18,0: PRINT "THE AUGMENTED SYS
TEM HAS REPEATED EIGENVALUES."
34030 LOCATE 14,19,0: PRINT "THE ALGORITHM WILL NOT WORK FOR THIS CAS
E.": GOSUB 30850: RS=0: RETURN
34040 NEXT J, I: FOR I=1 TO N: FOR J=1 TO N*2: CAA1(I,J)=0: CAA2(I,J)=0:
NEXT J, I
34050 FOR I=1 TO N: NI=0: FOR J=1 TO N: IF I=J THEN 34160
34055 DR=SPO(I,1)-SPO(J,1): DI=SPO(I,2)-SPO(J,2)
34060 FOR K1=1 TO N: FOR K2=1 TO N: IF K2<>K1 THEN 34080
34070 CAA1(K1,K2)=FNCRD(AA(K1,K2)-SPO(J,1),-SPO(J,2),DR,DI)
34075 CAA1(K1,K2+N)=FNCID(AA(K1,K2)-SPO(J,1),-SPO(J,2),DR,DI): GOTO
34090
34080 CAA1(K1,K2)=FNCRD(AA(K1,K2),0,DR,DI)
34085 CAA1(K1,K2+N)=FNCID(AA(K1,K2),0,DR,DI)
34090 NEXT K2,K1: NI=NI+1: IF NI<>1 THEN 34110
34100 FOR K1=1 TO N: FOR K2=1 TO N*2: CAA2(K1,K2)=CAA1(K1,K2): NEXT K2

```

```

,K1: GOTO 34160
34110 FOR K1=1 TO N: FOR K2=1 TO N: CAA3(K1,K2)=0: CAA3(K1,K2+N)=0: FOR
  K3=1 TO N
34115 R1=CAA2(K1,K3): AI1=CAA2(K1,K3+N): R2=CAA1(K3,K2): AI2=CAA1(K3,K
  2+N)
34120 CAA3(K1,K2)=CAA3(K1,K2)+FNCRM(R1,AI1,R2,AI2)
34130 CAA3(K1,K2+N)=CAA3(K1,K2+N)+FNCIM(R1,AI1,R2,AI2)
34140 NEXT K3,K2,K1: FOR K1=1 TO N: FOR K2=1 TO N*2: CAA2(K1,K2)=CAA3(
  K1,K2): NEXT K2,K1
34160 NEXT J: IF SPO(I,2)>.00001 THEN 34180
34170 FOR K1=1 TO N: FOR K2=1 TO N: CF(I,K1,K2)=CAA2(K1,K2): NEXT K2,K
  1: GOTO 34200
34180 IM=I+1: FOR K1=1 TO N: FOR K2=1 TO N: CF(I,K1,K2)=CAA2(K1,K2)*2
34190 CF(IM,K1,K2)=CAA2(K1,K2+N)*2: NEXT K2,K1: I=I+1
34200 NEXT I: RETURN
34999 '
35000 IF N=1 THEN SPO(N,1)=-SDI(N): SPO(N,2)=0: RETURN
35010 IF N=3 THEN 35070
35020 IF N=4 THEN 35230
35030 SR=SDI(2)*2-4*SDI(1): SI=0: N=2: GOSUB 35680
35040 IF RI<>0 THEN 35060 ELSE SPO(1,2)=0: SPO(2,2)=0
35050 SPO(1,1)=(-SDI(2)+RR)/2: SPO(2,1)=(-SDI(2)-RR)/2: RETURN
35060 SPO(1,1)=-SDI(2)/2: SPO(2,1)=SPO(1,1): SPO(1,2)=RI/2: SPO(2,2)=-
  RI/2: RETURN
35070 A=(3*SDI(2)-SDI(3)*2)/3: B=(2*SDI(3)*3-9*SDI(3)*SDI(2)+27*SDI(
  1))/27
35080 SR=B*2/4+A*3/27: SI=0: N=2: GOSUB 35680
35090 QTR=RR: QTI=RI: SR=QTR-B/2: SI=QTI: N=3
35100 IF SR<0 AND SI=0 THEN SR=-SR: GOSUB 35680: RR=-RR: GOTO 35120
35110 GOSUB 35680
35120 QAR=RR: QAI=RI: SR=-QTR-B/2: SI=-QTI
35130 IF SR<0 AND SI=0 THEN SR=-SR: GOSUB 35680: RR=-RR: GOTO 35150
35140 GOSUB 35680
35150 QBR=RR: QBI=RI: SR=-3: SI=0: N=2: GOSUB 35680
35160 SPO(1,1)=QAR+QBR: SPO(1,2)=QAI+QBI: QTR=(QAR-QBR)/2: QTI=(QAI-QB
  I)/2
35170 SR=FNCRM(QTR,QTI,RR,RI): SI=FNCIM(QTR,QTI,RR,RI)
35180 SSR=-(QAR+QBR)/2: SSI=-(QAI+QBI)/2: SPO(2,1)=SSR+SR: SPO(2,2)=SS
  I+SI: SPO(3,1)=SSR-SR: SPO(3,2)=SSI-SI
35190 FOR I=1 TO 3: SPO(I,1)=SPO(I,1)-SDI(3)/3: NEXT I
35200 MIN=ABS(SPO(1,2)): NX=1: FOR I=2 TO 3: IF ABS(SPO(I,2))<MIN THEN
  MIN=ABS(SPO(I,2)): NX=I
35210 NEXT I: IF NX<>1 THEN T=SPO(1,1): SPO(1,1)=SPO(NX,1): SPO(NX,1)=
  T: T=SPO(1,2): SPO(1,2)=SPO(NX,2): SPO(NX,2)=T
35220 IF SPO(2,2)<SPO(3,2) THEN RETURN ELSE T=SPO(2,2): SPO(2,2)=SPO
  (3,2): SPO(3,2)=T: RETURN
35230 A=SDI(4): B=SDI(3): C=SDI(2): D=SDI(1)
35240 FOR I=1 TO 4: SPO(I,1)=0: SPO(I,2)=0: NEXT I: PI=3.1415926#: PIO2=
  1.5707963#
35250 QP=-B: QQ=A*C-4*D: QR=-(A*2)*D+4*B*D-C*2
35260 QCA=(3*QQ-QP*2)/3: QCB=(2*QP*3-9*QP*QQ+27*QR)/27

```

```

35270 QTEMP=QCB/2/4+QCA/3/27:N=2:SR=QTEMP:SI=0
35280 GOSUB 35680:QTR=RR:QTI=RI
35290 SR=QTR-QCB/2:SI=QTI:N=3
35300 IF SR<0 AND SI=0 THEN SR=-SR:GOSUB 35680:RR=-RR:GOTO 35320
35310 GOSUB 35680
35320 QCAAR=RR:QCAAI=RI:SR=-QTR-QCB/2:SI=-QTI
35330 IF SR<0 AND SI=0 THEN SR=-SR:GOSUB 35680:RR=-RR:GOTO 35350
35340 GOSUB 35680
35350 YR=QCAAR+RR-QP/3:YI=QCAAI+RI
35360 SR=A/2/4-B+YR:SI=YI:N=2:GOSUB 35680
35370 QRR=RR:QRI=RI
35380 IF QRR=0 AND QRI=0 THEN 35480
35390 QTEMP=4*A*B-8*C-A/3
35400 QTR=FNCRD(QTEMP,0,4*QRR,4*QRI)
35410 QTI=FNCID(QTEMP,0,4*QRR,4*QRI)
35420 QRTR=FNCRM(QRR,QRI,QRR,QRI)
35430 QRTI=FNCIM(QRR,QRI,QRR,QRI)
35440 SR=3*A/2/4-QRTR-2*B+QTR:N=2:SI=-QRTI+QTI:GOSUB 35680
35450 QDR=RR:QDI=RI
35460 SR=3*A/2/4-QRTR-2*B-QTR:N=2:SI=-QRTI-QTI:GOSUB 35680
35470 QER=RR:QEI=RI:GOTO 35530
35480 QTR=FNCRM(YR,YI,YR,YI)-4*D
35490 QTI=FNCIM(YR,YI,YR,YI)
35500 SR=QTR:SI=QTI:N=2:GOSUB 35680:QTR=2*RR:QTI=2*RI
35510 SR=QTR+3*A/2/4-2*B:SI=QTI:GOSUB 35680:QDR=RR:QDI=RI
35520 SR=-QTR+3*A/2/4-2*B:SI=-QTI:GOSUB 35680:QER=RR:QEI=RI
35530 SP0(1,1)=-A/4+QRR/2+QDR/2:SP0(1,2)=QRI/2+QDI/2
35540 SP0(2,1)=-A/4+QRR/2-QDR/2:SP0(2,2)=QRI/2-QDI/2
35550 SP0(3,1)=-A/4-QRR/2+QER/2:SP0(3,2)=-QRI/2+QEI/2
35560 SP0(4,1)=-A/4-QRR/2-QER/2:SP0(4,2)=-QRI/2-QEI/2
35570 FOR I=1 TO 4:FOR J=I TO 4:IF ABS(SP0(J,2))>ABS(SP0(I,2)) THEN
35590
35580 T=SP0(I,1):SP0(I,1)=SP0(J,1):SP0(J,1)=T:T=SP0(I,2):SP0(I,2)=S
P0(J,2):SP0(J,2)=T
35590 NEXT J,I:IF SP0(1,2)<SP0(2,2) THEN 35610 ELSE T=SP0(1,1):SP0(
1,1)=SP0(2,1)
35600 SP0(2,1)=T:T=SP0(1,2):SP0(1,2)=SP0(2,2):SP0(2,2)=T
35610 IF SP0(3,2)<SP0(4,2) THEN 35630 ELSE T=SP0(3,1):SP0(3,1)=SP0(
4,1)
35620 SP0(4,1)=T:T=SP0(3,2):SP0(3,2)=SP0(4,2):SP0(4,2)=T
35630 IF SP0(1,2)-SP0(3,2)>.00001 THEN RETURN
35640 IF SP0(1,1)-SP0(1,2)<.00001 THEN RETURN ELSE T=SP0(2,1):SP0(2
,1)=SP0(4,1)
35650 SP0(4,1)=T:T=SP0(2,2):SP0(2,2)=SP0(4,2):SP0(4,2)=T:RETURN
35660 '
35670 'This subroutine finds the Nth root of a complex number.
35680 PI=3.1415926#:PIO2=1.5707963#
35690 IF SI=0 AND SR=0 THEN RI=0:RR=0:RETURN
35700 IF SI<0 AND SR<0 THEN ANGLE=-PI+ATN(SI/SR):GOTO 35780
35710 IF SI<0 AND SR=0 THEN ANGLE=-PIO2:GOTO 35780
35720 IF SI<0 AND SR>0 THEN ANGLE=-ATN(-SI/SR):GOTO 35780

```

```
35730 IF SI=0 AND SR<0 THEN ANGLE=PI: GOTO 35780
35740 IF SI=0 AND SR>0 THEN ANGLE=0: GOTO 35780
35750 IF SI>0 AND SR<0 THEN ANGLE=PI-ATN(-SI/SR): GOTO 35780
35760 IF SI>0 AND SR=0 THEN ANGLE=PI/2: GOTO 35780
35770 ANGLE=ATN(SI/SR)
35780 NR=SQR(SI2+SR2): RR=NR*(1/N)*COS(ANGLE/N): RI=NR*(1/N)*SIN(AN
GLE/N)
35790 RETURN
```

C.6 MAT 3/6/85

```

10 COMMON HELP$,HELP$,ARC%,SND%,QW,ARCNAM$: OPTION BASE 1:DEFINT I,J
20 MF$="-----MATRIX FUNCTIONS-----"
40 DIM A(20,20),B(20,20),C(21,20),D(21,20),SP1(20,20),SG(21,20)
50 DIM SX(20),SW(20,20),SP0(21,20),SDI(20,20),KDIM(5),SDJ(21,20)
52 DEF FNCRD(XR,XI,YR,YI)=(XR*YR+XI*YI)/(YR^2+YI^2)
54 DEF FNCID(XR,XI,YR,YI)=(XI*YR-XR*YI)/(YR^2+YI^2)
56 DEF FNCRM(XR,XI,YR,YI)=(XR*YR-XI*YI)
58 DEF FNCIM(XR,XI,YR,YI)=(XR*YI+XI*YR)
60 IF QW<1 THEN HELP%=0:HELP$="OFF":ARC%=0:SND%=1
70 MAXDIM=20:DIGIT$="1234567890":RT$="RETRY? (Y/N)"
90 EM=0:IF ARC%=1 THEN GOSUB 31710:COM$=MF$:GOSUB 30410:GOSUB 31710
100 CLS:WIDTH 80:COLOR 14,0:LOCATE 5,29,0:PRINT"MATRIX FUNCTIONS"
110 COLOR 10,0:LOCATE 8,23,0:PRINT"<F1>.....ENTER NEW MATRIX"
120 LOCATE 9,23,0:PRINT"<F2>.....RECALL OLD MATRIX"
130 LOCATE 10,23,0:PRINT"<F3>.....MODIFY CURRENT MATRIX"
140 LOCATE 11,23,0:PRINT"<F4>.....STORE MATRIX ON DISK"
150 LOCATE 12,23,0:PRINT"<F6>.....CALCULATE DATA"
160 LOCATE 13,23,0:PRINT"<F9>.....FILE DIRECTORY"
170 LOCATE 14,23,0:PRINT"<ESC>.....EXIT TO MAIN MENU"
180 LOCATE 15,23,0:PRINT"<Alt-F1>....HELP LEVEL"
190 LOCATE 16,23,0:PRINT"<Alt-F2>....SOUND LEVEL"
200 COLOR 14,0:LOCATE 15,46,0:PRINT HELP$:LOCATE 16,46,0:PRINT SND%
210 COLOR 10,0:LOCATE 17,23,0:PRINT"<Alt-F3>....ARCHIVE/COMMENTS"
220 COLOR 14,0:LOCATE 20,24,0:PRINT"PRESS DESIRED FUNCTION KEY"
230 INFLG=2:GOSUB 7120:IF ESCK=1 AND QW=1 THEN CHAIN"DACS"
235 IF ESCK=1 AND QW<>1 THEN 342
240 IF C=59 THEN GOSUB 19190:GOTO 100
250 IF C=60 THEN GOSUB 19530:GOTO 100
260 IF C=61 THEN GOSUB 19790:GOTO 100
270 IF C=62 THEN GOSUB 19330:GOTO 100
280 IF C=64 THEN GOSUB 19840:GOTO 100
290 IF C=67 THEN GOSUB 30190:GOTO 100
300 IF C=104 THEN GOSUB 590:GOTO 200
310 IF C=105 THEN GOSUB 550:GOTO 200
320 IF C=106 THEN GOSUB 360:GOTO 100
330 GOSUB 30760:GOTO 230
340 '
341 ' SYSTEM
342 CLEAR:WIDTH 80:SCREEN 0,0:COLOR 7,0:CLOSE:END
343 '
350 ' ARCHIVE
360 IF ARC%=1 THEN GOSUB 420:RETURN
370 GOSUB 420:IF ARC%=0 THEN RETURN ELSE GOSUB 31710:COM$=MF$:GOSUB
30410:GOSUB 31710:RETURN
400 '
410 ' ARCHIVE SESSION / ADD COMMENTS
420 COLOR 12,0:CLS:IF ARC%=1 THEN 470 ELSE LOCATE 2,27,0
430 PRINT"ARCHIVE SESSION? (Y/N)":INFLG=1:GOSUB 7120:IF ESCK=1 OR
C$="N" THEN RETURN

```

```

440 FILETPE=0: IF HELP%=1 THEN GOSUB 30520 ELSE GOSUB 30580
450 FILEXT$=".ARC": FILENUM=2: GOSUB 30650: CLOSE #FILENUM
460 IF ESCK=1 OR FERR$<>"OK" THEN RETURN ELSE ARCNAM$=FILENAM$: ARC%
=1: CLS
470 COLOR 12,0: LOCATE 2,28,0: PRINT"ENTER COMMENTS? (Y/N)": INFLG=1:
GOSUB 7120
480 IF C$="N" OR ESCK=1 THEN RETURN ELSE NUMDIG=78: INFLG=5: XFLD=1
490 CLS: LOCATE 2,19,0: YFLD=4
500 PRINT"ENTER COMMENTS BELOW....PRESS ESC TO QUIT": PRINT
510 GOSUB 7110: IF ESCK=1 THEN GOSUB 31710: RETURN
520 COM$=S$: GOSUB 30410: IF FERR$<>"OK" THEN RETURN ELSE YFLD=YFLD+1
: XFLD=1: IF YFLD>24 THEN 490 ELSE 510
530 '
540 'SOUND LEVEL
550 SND%=SND%+1: IF SND%>2 THEN SND%=0
560 RETURN
570 '
580 'SET HELP LEVEL
590 IF HELP%=1 THEN HELP%=0: HELP$="OFF": RETURN
600 HELP%=1: HELP$="ON ": RETURN
610 '
7040 'INPUT ROUTINE for numbers, names, function keys, comments, &
Y/N answers
7050 'INFLG=1: gets Y/N answer, returns c$="Y" or c$="N"
7060 'INFLG=2: gets function key, returns c=asc(2nd char of key)
7070 'INFLG=3: gets number as string, checks if valid, returns s=n
umber
7080 'INFLG=4: gets filename, returns s$=filename
7090 'INFLG=5: gets comment from user for archive, returns s$=comm
ent
7100 'Calling routine specifies NUMDIG, XFLD, YFLD, and INFLG
7110 COLOR 6,0
7120 SXFLD=XFLD: S$="": C$="": T$="": N$="": ESCK=0: GOSUB 30030: IF SND%>
1 THEN BEEP
7130 IF INFLG>2 THEN 7150
7140 C$=INKEY$: IF C$="" THEN 7140 ELSE 7170
7150 LOCATE YFLD,XFLD,1
7160 C$=INKEY$: IF C$="" THEN 7150
7170 IF C$=CHR$(27) THEN ESCK=1: RETURN 'escape key
7180 IF INFLG=2 THEN 7370
7190 IF LEN(C$)>1 THEN 7450 'invalid
7200 C=ASC(C$)
7210 IF INFLG=1 THEN 7320
7220 IF C=8 THEN 7500
7230 IF INFLG=5 THEN 7800 'comment
7240 IF INFLG=3 THEN 7390 'number
7250 'Name Routine
7260 IF C=13 THEN RETURN
7270 IF C=46 AND NUMDIG>10 THEN 7470 'okay
7280 IF C=91 OR C=92 OR C=93 OR C=124 THEN 7450 'invalid
7290 IF C=34 OR C=39 OR C=42 OR C=43 OR C=46 OR C=47 THEN 7450 'in

```

valid

```

7300 IF (C>32 AND C<59) OR (C>63 AND C<127) THEN 7470 ELSE 7450
7310 'Yes/No routine
7320 IF C=89 OR C=121 THEN C$="Y":RETURN
7330 IF C=78 OR C=110 THEN C$="N":RETURN
7340 IF C=13 THEN C$="N":RETURN
7350 GOTO 7450
7360 'Function Key
7370 IF LEN(C$)<2 THEN 7450 ELSE C$=RIGHT$(C$,1):C=ASC(C$)
7380 IF (C>58 AND C<69) OR (C>83 AND C<114) THEN RETURN ELSE 7450
7390 'Number Routine
7400 IF C=13 THEN 7540 'carriage return
7410 IF C>47 AND C<58 THEN 7470 'okay
7420 IF C=46 OR C=43 OR C=45 OR C=32 THEN 7470 'okay
7430 IF C=69 OR C=101 THEN C$=CHR$(69):GOTO 7470 'okay
7440 'Illegal character
7450 GOSUB 30760:GOTO 7130
7460 'Character okay
7470 IF LEN(S$)>=NUMDIG THEN 7450
7480 LOCATE YFLD,XFLD:PRINT C$;:S$=S$+C$:XFLD=XFLD+1:GOTO 7130
7490 'Backspace
7500 XFLD=XFLD-1:LOCATE YFLD,XFLD:PRINT" ";:IF XFLD<=SXFLD THEN XFLD=SXFLD
7510 LNT=LEN(S$)-1:IF LNT<0 THEN LNT=0
7520 S$=MID$(S$,1,LNT):GOTO 7130
7530 'Number Check routine
7540 NUMCK=0:DECPT=0:EXPNT=0:SIGN=0
7550 IF S$="" THEN 7770 ELSE T$=LEFT$(S$,1)
7560 IF T$="+" OR T$="-" THEN N$=T$:S$=MID$(S$,2)
7570 IF T$="." THEN N$=T$:DECPT=1:S$=MID$(S$,2)
7580 IF S$="" THEN 7790
7590 IF (T$="+" OR T$="-") AND INSTR(".",LEFT$(S$,1))=1 THEN DECPT=1:T$=LEFT$(S$,1):N$=N$+T$:S$=MID$(S$,2)
7600 IF INSTR(DIGIT$,LEFT$(S$,1))=0 THEN 7790
7610 WHILE S$<>"":T$=LEFT$(S$,1):S$=MID$(S$,2)
7620 IF T$=" " THEN 7680
7630 IF (T$="+" OR T$="-") AND RIGHT$(N$,1)<>"E" THEN NUMCK=1:GOTO 7680
7640 IF T$="." THEN IF DECPT=1 OR EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:DECPT=1
7650 IF T$="+" OR T$="-" THEN IF SIGN=1 OR EXPNT=0 THEN NUMCK=1 ELSE N$=N$+T$:SIGN=1
7660 IF T$="E" THEN IF EXPNT=1 THEN NUMCK=1 ELSE N$=N$+T$:EXPNT=1:DECPT=1
7670 IF T$<> "." AND T$<> "+" AND T$<> "-" AND T$<> "E" THEN N$=N$+T$
7680 WEND:IF NUMCK=1 THEN 7790
7690 'Check for overflow
7700 D=0:N%=INSTR(N$,"E"):IF N%=0 THEN 7770
7710 IF N%=1 THEN 7790
7720 S$=LEFT$(N$,N%-1):C$=MID$(N$,N%+1):S=VAL(S$)
7730 IF S=0 THEN D=0 ELSE D=INT(LOG(ABS(S))/LOG(10))

```

```

7740 IF C$<>"" THEN D=D+VAL(C$)
7750 IF D<-37 OR D>37 THEN 7790
7760 'convert
7770 S=VAL(N$):RETURN
7780 'invalid
7790 NUMCK=1:GOSUB 30760:BEEP:LOCATE YFLD,SXFLD,0:PRINT SPC(NUMDIG)
: XFLD=SXFLD:GOTO 7120
7800 'Comment routine
7810 IF C$=CHR$(13) THEN RETURN
7820 IF LEN(S$)>=NUMDIG THEN RETURN ELSE 7480
7830 '
19180 'ENTER NEW MATRIX ROUTINE
19190 CLS: IF EM=0 THEN 19220 ELSE COM$="ERASE EXISTING MATRIX? (Y/
N)":GOSUB 32340
19200 IF C$="N" OR ESCK=1 THEN RETURN ELSE CLS
19220 COLOR 12,0:LOCATE 10,29,0:PRINT"MATRIX DIMENSION >"SPC(15)
19230 XFLD=49:YFLD=10:NUMDIG=12:INFLG=3:GOSUB 7110:IF ESCK=1 THEN 1
9290
19240 S=INT(S):IF S<=MAXDIM AND S>0 THEN KDIM(1)=S:GOTO 19260
19250 GOSUB 30760:COLOR 15,0:LOCATE 12,18,0:PRINT"ERROR! Value mus
t lie in the range 1 to"MAXDIM:GOTO 19220
19260 NUMDIG=10:INFLG=3:P=0:CLS:IF HELP%=0 THEN 19280 ELSE COLOR 3,
0
19270 LOCATE 1,6,0:PRINT"ENTER MATRIX BELOW BY ROWS PRESSING <ENTER
> AFTER EACH ELEMENT."
19280 MAT=1:COM$="MATRIX":ROW%=KDIM(1):COL%=KDIM(1):XFLD=1:YFLD=3:G
OSUB 30890:IF ESCK=1 THEN 19290 ELSE EM=1
19285 IF ARC%=1 THEN GOSUB 31710:COM$="-ENTER NEW MATRIX-":GOSUB 30
410
19288 RETURN
19290 COM$="MATRIX NOT ENTERED!!":GOSUB 32310:RETURN
19310 '
19320 'SAVE MATRIX ROUTINE
19330 IF EM=0 THEN COM$="NO MATRIX TO SAVE!!":GOSUB 32310:RETURN
19360 FILETPE=0:IF HELP%=1 THEN GOSUB 30520 ELSE CLS:GOSUB 30580
19370 FILEXT$=".MAT":FILENUM=1:GOSUB 30650
19380 IF ESCK=0 AND FERR$="OK" THEN 19410
19390 COM$="MATRIX NOT SAVED!!":GOSUB 32310:RETURN
19410 SFILE$=FILENAM$:CLS:COLOR 3,0:LOCATE 10,30,0:PRINT"SAVING MAT
RIX....."
19420 IF ARC%=1 THEN GOSUB 31710:COM$="-STORE MATRIX ON DISK-":GOSU
B 30410:FILENUM=1
19430 MAT=0:ROW%=1:COL%=1:GOSUB 31380:IF FERR$<>"OK" THEN 19470
19440 MAT=1:ROW%=KDIM(1):COL%=ROW%:GOSUB 31380:IF FERR$<>"OK" THEN
19470
19450 CLOSE #FILENUM:IF ARC%=1 THEN COM$="Matrix saved under filena
me "+SFILE$:GOSUB 30410
19460 RETURN
19470 COM$=FERR$:GOSUB 32700:FILENAM$=SFILE$:GOSUB 31530:IF C$="Y"
AND ESCK=0 THEN 19360
19480 IF ARC%=1 THEN COM$=FERR$:GOSUB 19360

```

```

19490 GOTO 19390
19510 '
19520 'RECALL MATRIX ROUTINE
19530 IF EM=1 THEN COM$='ERASE EXISTING MATRIX? (Y/N)':GOSUB 32340
: IF C$="N" OR ESCK=1 THEN RETURN
19560 FILETPE=1: IF HELP%=1 THEN GOSUB 30520 ELSE CLS:GOSUB 30580
19570 FILEXT$=".MAT": FILENUM=1:GOSUB 30650
19580 IF ESCK=0 AND FERR$="OK" THEN SFILE$=FILENAM$:GOTO 19610
19590 COM$="MATRIX NOT RECALLED!!":GOSUB 32310: RETURN
19610 CLS: COLOR 3,0: LOCATE 10,30,0: PRINT"RECALLING MATRIX....."
19620 IF ARC%=1 THEN GOSUB 31710: COM$="-RECALL MATRIX-":GOSUB 30410
: FILENUM=1
19630 ROW%=1: COL%=1: MAT=0:GOSUB 31110: IF FERR$<>"OK" THEN 19720
19640 ROW%=KDIM(1): COL%=ROW%: MAT=1:GOSUB 31110: IF FERR$<>"OK" THEN
19720
19650 EM=1: CLOSE #FILENUM: COLOR 7,0: LOCATE 14,30,0: PRINT"SEE MATRIX
? (Y/N)"
19660 INFLG=1:GOSUB 7120: IF C$="Y" AND ESCK=0 THEN 19690
19670 IF ARC%=1 THEN COM$="Matrix recalled from file "+SFILE$:GOSUB
30410
19680 RETURN
19690 CLS: COLOR 12,0: LOCATE 10,29,0: PRINT"SYSTEM DIMENSION > ": COLO
R 6,0: LOCATE 10,48,0: PRINT KDIM(1)
19700 GOSUB 30850: IF ESCK=1 THEN 19670 ELSE P=1: NUMDIG=10: CLS
19710 YFLD=3: XFLD=1: MAT=1: COM$="MATRIX": ROW%=KDIM(1): COL%=ROW%: GOSU
B 30890:GOSUB 30850:GOTO 19670
19720 CLOSE #FILENUM: COM$=FERR$:GOSUB 32700: IF C$="Y" AND ESCK=0 TH
EN 19560
19750 IF ARC%=1 THEN COM$=FERR$+" for file "+SFILE$:GOSUB 30410
19760 GOTO 19590
19770 '
19780 'MODIFY MATRIX ROUTINE
19790 IF EM=0 THEN COM$="NO MATRIX TO MODIFY!!":GOSUB 32310: RETURN
19800 IF ARC%=1 THEN GOSUB 31710: COM$="-MODIFY CURRENT MATRIX-":GOS
UB 30410
19810 COM$="MATRIX": MAT=1: ROW%=KDIM(1): COL%=ROW%:GOSUB 32440: RETURN
19820 '
19830 'CALCULATE MATRIX DATA
19840 IF EM=0 THEN COM$="ENTER MATRIX FIRST!!":GOSUB 32310: RETURN
19850 IF ARC%=1 THEN GOSUB 31710: COM$="-CALCULATE MATRIX DATA-":GOS
UB 30410
19860 GOSUB 32610
19870 INFLG=2:GOSUB 7120: IF ESCK=1 THEN RETURN
19880 IF C=84 THEN GOSUB 20010:GOTO 19860
19890 IF C=85 THEN GOSUB 20460:GOTO 19860
19900 IF C=86 THEN GOSUB 20210:GOTO 19860
19910 IF C=87 THEN GOSUB 20660:GOTO 19860
19920 IF C=88 THEN GOSUB 19970:GOTO 19860
19930 IF C=89 THEN GOSUB 21060:GOTO 19860
19940 GOSUB 30760:GOTO 19870
19950 '

```

```

' det
' inv
' char poly
' eigen
' phi
' resolve

```

```

19960 'CALCULATE STATE TRANSITION MATRIX
19970 IF ARC%=1 THEN COM$="Calculated State Transition Matrix":GOSU
B 30410
19972 CLS:COLOR 12,0:LOCATE 10,31,0:PRINT"TIME STEP >"SPC(38):XFLD=
43:YFLD=10
19973 NUMDIG=12:INFLG=3:GOSUB 7110:IF ESCK=1 THEN RETURN ELSE T=S
19974 COLOR 3,0:LOCATE 12,29,0:PRINT"CALCULATING DATA....."
19975 RS=1:GOSUB 31740:IF RS=0 THEN RETURN ELSE MAT=6:XFLD=1:YFLD=3
:CLS:P=1
19980 NUMDIG=12:ROW%=KDIM(1):COL%=ROW%:COM$="STATE TRANSITION MATRI
X":GOSUB 30890:GOSUB 30850:RETURN
19990 '
20000 'CALCULATE DETERMINANT
20010 IF ARC%=1 THEN COM$="Calculated Determinant":GOSUB 30410
20015 NN%=KDIM(1):FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1):B(I,J)=A(I,
J):NEXT J,I
20016 GOSUB 20020:CLS:COLOR 12,0:LOCATE 10,32,0:PRINT"DETERMINANT =
":COLOR 6,0
20017 LOCATE 10,46,0:PRINT DET;:GOSUB 30850:RETURN
20020 IREV=0:FOR I=1 TO NN%:KKK%=I
20030 IF B(KKK%,I)<>0 THEN 20050 ELSE KKK%=KKK%+1
20040 IF KKK%<=NN% THEN 20030 ELSE 20130
20050 IF I<KKK% THEN 20070
20060 IF I>KKK% THEN 20130 ELSE 20080
20070 FOR JJ=1 TO NN%:TEMP=B(I,JJ):B(I,JJ)=B(KKK%,JJ):B(KKK%,JJ)=TE
MP:NEXT JJ:IREV=IREV+1
20080 II=I+1:IF II>NN% THEN 20110
20090 FOR JJ=II TO NN%:IF B(JJ,I)=0 THEN 20105 ELSE TEMP=B(JJ,I)/B(
I,I)
20100 FOR N%=I TO NN%:B(JJ,N%)=B(JJ,N%)-B(I,N%)*TEMP:NEXT N%
20105 NEXT JJ,I
20110 DET=1:FOR I=1 TO NN%:DET=DET*B(I,I):NEXT I
20120 DET=(-1)IREV*DET:RETURN
20130 DET=0:RETURN
20190 '
20200 'CALCULATE CHARACTERISTIC POLYNOMIAL
20210 CLS:EIGEN=0:COLOR 3,0:IF HELP%=0 THEN 20240 ELSE LOCATE 5,11,
0
20220 PRINT"The following routine calculates the characteristic pol
ynomial"
20230 LOCATE 6,35,0:PRINT"det(sI-A)"
20240 LOCATE 9,29,0:PRINT"CALCULATING DATA....."
20245 IF ARC%=1 THEN COM$="Calculated Characteristic Polynomial":GO
SUB 30410
20246 GOSUB 20260:IF RS=0 OR EIGEN=1 THEN RETURN ELSE COLOR 12,0
20248 LOCATE 12,13,0:PRINT"The Characteristic Polynomial in ascendi
ng powers of s"
20250 COLOR 6,0:LOCATE 13,(38-3*KDIM(1)/2),0:FOR I=1 TO KDIM(1)+1:P
RINT C(I,1);:NEXT I:GOSUB 30850:RETURN
20260 RS=1:FOR I=1 TO KDIM(1):C(I,1)=0:NEXT I:C(KDIM(1)+1,1)=1
20265 FOR J=1 TO KDIM(1):K%=0:L%=1:D(1,1)=0

```

```

20270 D(L%,1)=D(L%,1)+1: IF (L%-J)<0 THEN 20290
20280 IF (L%-J)>0 THEN 20380 ELSE 20300
20290 FOR I=L% TO J-1: D(I+1,1)=D(I,1)+1: NEXT I
20300 FOR I=1 TO J: FOR KK%=1 TO J: ROW%=D(I,1): COL%=D(KK%,1): B(I, KK%)
)=A(ROW%, COL%): NEXT KK%, I
20310 K%=K%+1: NN%=J: GOSUB 20020: SX(K%)=DET
20320 FOR I=1 TO J: L%=J-I+1: IF (D(L%,1)-(KDIM(1)-J+L%))<0 THEN 2027
0
20330 IF (D(L%,1)-(KDIM(1)-J+L%))>0 THEN 20380
20340 NEXT I: S%=KDIM(1)-J+1
20350 FOR I=1 TO K%: C(S%,1)=C(S%,1)+SX(I)*(-1)^J: NEXT I, J: RETURN
20380 CLS: GOSUB 30760: COLOR 15,0: LOCATE 10,28,0: PRINT "ERROR IN CALC
ULATIONS!!"
20390 COLOR 7,0: LOCATE 11,26,0: PRINT "Please re-evaluate matrix": GOS
UB 30850
20400 IF ARC%=1 THEN COM$="ERROR CALCULATING CHARACTERISTIC POLYNOM
IAL": GOSUB 30410: GOSUB 30710
20410 RS=0: RETURN
20440 '
20450 ' CALCULATE INVERSE
20460 IF ARC%=1 THEN COM$="Calculated Inverse": GOSUB 30410
20470 FOR I=1 TO KDIM(1): FOR J=1 TO KDIM(1): B(I,J)=A(I,J): IF I=J TH
EN C(I,J)=1 ELSE C(I,J)=0
20480 NEXT J, I: N%=1: FOR I=1 TO KDIM(1): S=0: K%=I
20485 IF (ABS(B(K%,I))-ABS(S))<=0 THEN 20500
20490 S=B(K%,I): N%=K%
20500 K%=K%+1: IF (K%-KDIM(1))<=0 THEN 20485
20510 IF B(N%,I)=0 THEN 20590
20520 IF (N%-I)<0 THEN 20590 ELSE IF (N%-I)=0 THEN 20540
20530 FOR J=1 TO KDIM(1): TEMP=B(I,J): B(I,J)=B(N%,J): B(N%,J)=TEMP: TE
MP=C(I,J): C(I,J)=C(N%,J): C(N%,J)=TEMP: NEXT J
20540 TEMP=B(I,I): FOR J=1 TO KDIM(1): C(I,J)=C(I,J)/TEMP: B(I,J)=B(I,
J)/TEMP: NEXT J
20550 FOR J=1 TO KDIM(1): IF (J-I)=0 THEN 20575
20555 IF B(J,I)=0 THEN 20575 ELSE TEMP=B(J,I)
20560 FOR N%=1 TO KDIM(1): C(J,N%)=C(J,N%)-TEMP*C(I,N%)
20570 B(J,N%)=B(J,N%)-TEMP*B(I,N%): NEXT N%
20575 NEXT J, I
20580 CLS: MAT=3: XFLD=1: YFLD=3: P=1: NUMDIG=12: COM$="INVERSE": ROW%=KDI
M(1): COL%=ROW%: GOSUB 30890: GOSUB 30850: RETURN
20590 CLS: COLOR 15,0: GOSUB 30760: LOCATE 10,29,0: PRINT "MATRIX IS SIN
GULAR!!"
20600 IF ARC%=1 THEN COM$="MATRIX SINGULAR!!": GOSUB 30410: GOSUB 317
10
20610 GOSUB 30850: RETURN
20640 '
20650 ' CALCULATE EIGENVALUES
20660 IF ARC%=1 THEN COM$="Calculated Eigenvalues": GOSUB 30410
20665 CLS: COLOR 3,0: LOCATE 10,29,0: PRINT "CALCULATING DATA....."
20670 EIGEN=1: GOSUB 20260: IF RS=0 THEN RETURN
20680 IREV=1: NC=KDIM(1)+1: FOR I=1 TO NC: D(I,1)=C(I,1): NEXT I: P=0: Q=

```

```

0: R=0
20685 N=KDIM(1): IF N<5 THEN GOSUB 33000: GOTO 21000
20690 IF D(1,1)<>0 THEN 20710 ELSE NC=NC-1
20700 SPO(NC,1)=0: SPO(NC,2)=0: FOR I=1 TO NC: D(I,1)=D(I+1,1): NEXT I:
GOTO 20690
20710 IF (NC-1)=0 THEN 21000
20720 IF (NC-2)=0 THEN R=-D(1,1)/D(2,1): GOTO 20940
20730 IF (NC-3)=0 THEN P=D(2,1)/D(3,1): Q=D(1,1)/D(3,1): GOTO 20950
20740 IF (ABS(D(NC-1,1)/D(NC,1))-ABS(D(2,1)/D(1,1)))>=0 THEN 20790
20750 IREV=-IREV: FOR I=1 TO NC/2: NL=NC+1-I: TEMP=D(NL,1): D(NL,1)=D(I,1): D(I,1)=TEMP: NEXT I
20760 IF Q=0 THEN P=0: GOTO 20780
20770 P=P/Q: Q=1/Q
20780 IF R<>0 THEN R=1/R
20790 E=5E-10: B(NC,1)=D(NC,1): C(NC,1)=D(NC,1): B(NC+1,1)=0: C(NC+1,1)=0: NP=NC-1
20800 FOR J=1 TO 1000: FOR II=1 TO NP: I=NC-II: B(I,1)=D(I,1)+R*B(I+1,1)
20810 C(I,1)=B(I,1)+R*C(I+1,1): NEXT II
20820 IF (ABS(B(1,1)/D(1,1))-E)<=0 THEN 20940
20830 IF C(2,1)=0 THEN R=R+1 ELSE R=R-B(1,1)/C(2,1)
20840 FOR II=1 TO NP: I=NC-II
20850 B(I,1)=D(I,1)-P*B(I+1,1)-Q*B(I+2,1): C(I,1)=B(I,1)-P*C(I+1,1)-Q*C(I+2,1): NEXT II
20860 IF D(2,1)<>0 THEN 20880
20870 IF ABS(B(2,1)/D(1,1))-E<=0 THEN 20890 ELSE 20900
20880 IF ABS(B(2,1)/D(2,1))-E>0 THEN 20900
20890 IF ABS(B(1,1)/D(1,1))-E<=0 THEN 20950
20900 CBAR=C(2,1)-B(2,1): D=C(3,1)*2-CBAR*C(4,1)
20910 IF D=0 THEN P=P-2: Q=Q*(Q+1): GOTO 20930
20920 P=P+(B(2,1)*C(3,1)-B(1,1)*C(4,1))/D: Q=Q+(-B(2,1)*CBAR+B(1,1)*C(3,1))/D
20930 NEXT J: E=E*10: GOTO 20800
20940 NC=NC-1: SPO(NC,2)=0: IF IREV<0 THEN SPO(NC,1)=1/R ELSE SPO(NC,1)=R
20945 FOR I=1 TO NC: D(I,1)=B(I+1,1): NEXT I: GOTO 20710
20950 NC=NC-2: IF IREV<0 THEN QP=1/Q: PP=P/(Q*2): GOTO 20957
20955 QP=Q: PP=P/2
20957 TEMP=(PP)*2-QP: IF TEMP>=0 THEN 20970
20960 SPO(NC+1,1)=-PP: SPO(NC,1)=-PP: SPO(NC+1,2)=SQR(-TEMP): SPO(NC,2)=-SPO(NC+1,2): GOTO 20990
20970 IF PP=0 THEN SPO(NC+1,1)=-SQR(TEMP) ELSE SPO(NC+1,1)=(-PP/ABS(PP))*(ABS(PP)+SQR(TEMP))
20980 SPO(NC+1,2)=0: SPO(NC,1)=QP/SPO(NC+1,1): SPO(NC,2)=0
20990 FOR I=1 TO NC: D(I,1)=B(I+2,1): NEXT I: GOTO 20710
21000 CLS: COLOR 12,0: LOCATE 1,1,0: PRINT "The Eigenvalues of the matrix are"
21010 COM$="          Real Part  Imaginary Part": P=1: NUMDIG=12: ROW%=KDIM(1): COL%=2
21020 MAT=6: XFLD=1: YFLD=3: GOSUB 30890: GOSUB 30850: RETURN
21030 '

```

```

21050 ' CALCULATE RESOLVENT MATRIX
21060 IF ARC%=1 THEN COM$="Calculated Resolvent Matrix":GOSUB 30410
21070 CLS: COLOR 3,0: LOCATE 10,29,0: PRINT"CALCULATING DATA....."
21080 EIGEN=1:GOSUB 20260: IF RS=0 THEN RETURN
21090 FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1): IF I=J THEN SPO(I,J)=1
ELSE SPO(I,J)=0
21100 NEXT J,I:NNN=KDIM(1)-1:GOSUB 21200
21120 FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1): SPO(I,J)=A(I,J):NEXT J,
I
21130 FOR II=1 TO KDIM(1):NNN=KDIM(1)-II: IF II=1 THEN 21180 ELSE GO
SUB 21200
21140 IF II=KDIM(1) THEN 21190
21150 FOR J=1 TO KDIM(1):FOR K%=1 TO KDIM(1):D(J,K%)=0
21160 FOR L%=1 TO KDIM(1):D(J,K%)=D(J,K%)+(A(J,L%)*SPO(L%,K%)):NEXT
L%,K%,J
21170 FOR J=1 TO KDIM(1):FOR K%=1 TO KDIM(1): SPO(J,K%)=D(J,K%):NEXT
K%,J
21180 FOR J=1 TO KDIM(1): SPO(J,J)=SPO(J,J)+C(KDIM(1)-II+1,1):NEXT J
21190 NEXT II:RETURN
21200 CLS: COLOR 12,0: PRINT"The Matrix Coefficients of the Numerator
of the Resolvent Matrix"
21210 MAT=6: XFID=1: YFID=3: P=1: NUMDIG=12: ROW%=KDIM(1): COL%=KDIM(1)
21220 S$=STR$(NNN):COM$="MATRIX COEFFICIENT OF S"+S$:GOSUB 30890:G
OSUB 30850
21230 IF II<>KDIM(1) THEN CLS: COLOR 3,0: LOCATE 10,29,0: PRINT"CALCUL
ATING DATA....."
21240 RETURN
30000 '***** SUBROUTINES *****
30010 '
30020 ' CLEAR KBD BUFFER
30030 DEF SEG=0: POKE 1050,PEEK(1052):DEF SEG
30040 RETURN
30050 '
30060 ' OPEN DISK FILE FOR INPUT/OUTPUT
30070 ' Calling routine specifies filename$, filenum, fileapp, & file
tpe
30080 ' fileapp=0 no append, fileapp=1 append, filetype=0 output =1 i
nput
30090 ON ERROR GOTO 30150: IF FILEAPP=0 THEN 30110 ELSE FILEAPP=0
30100 OPEN FILENAM$ FOR APPEND AS #FILENUM:GOTO 30140
30110 IF FILETPE=1 THEN 30130
30120 OPEN FILENAM$ FOR OUTPUT AS #FILENUM:GOTO 30140
30130 OPEN FILENAM$ FOR INPUT AS #FILENUM
30140 FERR$="OK"
30145 ON ERROR GOTO 0:RETURN
30150 GOSUB 31260:RESUME 30145
30170 '
30180 ' FILE DIRECTORY SUBROUTINE
30190 ON ERROR GOTO 30280
30200 COLOR 12,0:CLS:GOSUB 30030:LOCATE 1,18,0: IF SND%>1 THEN BEEP
30210 PRINT"WHICH DRIVE? (A, B, OR AB for both) >";:COLOR 6,0:INPU

```

```

T"" ,COM$
30220 PRINT:COLOR 3,0
30230 IF COM$='a:' OR COM$='A:' OR COM$='a' OR COM$='A' THEN PRINT'
A:\":FILES"A:*. *':GOTO 30310
30240 IF COM$='b:' OR COM$='B:' OR COM$='b' OR COM$='B' THEN PRINT'
B:\":FILES"B:*. *':GOTO 30310
30250 IF COM$='ab:' OR COM$='AB:' OR COM$='ab' OR COM$='AB' THEN 30
260 ELSE 30270
30260 PRINT'A:\":FILES"A:*. *':PRINT'B:\":FILES"B:*. *':GOTO 30310
30270 GOSUB 30760:GOTO 30200
30280 GOSUB 31260:GOSUB 30760:COLOR 15,0:PRINT:PRINT SPC(30) FERR$:
COLOR 7,0
30290 PRINT:PRINT SPC(31) RT$:INFLG=1:GOSUB 7120
30300 IF C$='Y' THEN RESUME 30200 ELSE RESUME 30310
30310 ON ERROR GOTO 0
30320 YFLD=24:GOSUB 31680:LOCATE 24,28,0:COLOR 12,0:PRINT'KILL ANY
FILES? (Y/N)";:INFLG=1:GOSUB 7120:IF C$='N' OR ESCK=1 THEN RETURN
30330 GOSUB 31680:LOCATE 24,32,0:PRINT'FILENAME >";:COLOR 6,0
30340 YFLD=24:XFLD=43:NUMDIG=14:INFLG=4:GOSUB 7110:IF ESCK=1 THEN R
ETURN
30350 FILENAM$=S$:GOSUB 31530:IF FERR$='OK' THEN 30320 ELSE GOSUB 3
1680:GOSUB 30760
30360 COLOR 15,0:LOCATE 24,13,0:PRINT FERR$;:LOCATE 24,47,0:COLOR 7
,0:PRINT RT$;
30370 INFLG=1:GOSUB 7120:IF C$='N' OR ESCK=1 THEN RETURN ELSE COLOR
12,0:GOTO 30330
30380 '
30390 'PROGRAM WRITE TO ARCHIVE
30400 'Calling routine provides COM$
30410 FILENAM$=ARCNAM$:FILENUM=2:FILEAPP=1:FILETPE=0
30420 GOSUB 30090:IF FERR$<>"OK" THEN CLOSE #FILENUM:GOTO 30450 'op
en file
30430 ON ERROR GOTO 30490:PRINT #FILENUM,COM$
30440 CLOSE #FILENUM:ON ERROR GOTO 0:RETURN
30450 CLS:COLOR 15,0:LOCATE 10,28,0:PRINT'ERROR ARCHIVING SESSION'
30460 LOCATE 12,28,0:PRINT FERR$:COLOR 7,0:GOSUB 30760
30470 LOCATE 14,30,0:PRINT RT$:INFLG=1:GOSUB 7120
30480 IF C$='Y' THEN 30420 ELSE RETURN
30490 ON ERROR GOTO 0:GOSUB 31260:RESUME 30450
30500 '
30510 'FILENAME INFORMATION & DIRECTORY ROUTINE
30520 COLOR 3,0:CLS
30530 LOCATE 2,15,0:PRINT'A filename is 1-8 letters/digits. A file
name may"
30540 LOCATE 3,15,0:PRINT"be preceded by a disk drive letter and a
colon,"
30550 LOCATE 4,15,0:PRINT"otherwise the current logged drive is use
d."
30560 IF FILETPE=0 THEN COLOR 15,0:LOCATE 6,8,0:PRINT'WARNING! Usi
ng an existing filename will erase the existing file."
30570 COLOR 3,0:LOCATE 8,24,0:PRINT'Would you like to see your"

```

```

30580 COLOR 12,0:LOCATE 9,26,0:PRINT"File Directory? (Y/N)"
30590 INFLG=1:GOSUB 7120:IF C$="Y" THEN GOSUB 30190
30600 RETURN
30610 '
30620 'INPUT FILENAME ROUTINE
30630 'calling routine supplies filext$, filenum, filetype, fileapp
30640 'leaves file open if OK, closes if FO=1
30650 CLS:COLOR 3,0:LOCATE 2,24,0:PRINT"Enter filename including dr
ive."
30660 LOCATE 3,20,0:PRINT"File will be given the extension "FILEXT$
30670 LOCATE 5,30,0:COLOR 12,0:PRINT"FILENAME >"
30680 XFLD=41:YFLD=5:NUMDIG=10:INFLG=4:GOSUB 7110:IF ESCK=1 THEN RE
TURN
30690 FILENAM$=S$+FILEXT$:GOSUB 30090
30700 IF FERR$="OK" THEN RETURN ELSE CLOSE #FILENUM
30710 COLOR 15,0:GOSUB 30760:LOCATE 7,28,0:PRINT FERR$:COLOR 7,0
30720 LOCATE 9,30,0:PRINT RT$:INFLG=1:GOSUB 7120
30730 IF C$="Y" THEN 30650 ELSE RETURN
30740 '
30750 'FLUB SOUND SUBROUTINE
30760 IF SND%<1 THEN RETURN ELSE SOUND 1000,2:SOUND 1180,3:BEEP:RET
URN
30770 '
30780 'PRESS ANY KEY ROUTINE
30790 'Caller specifies XFLD & YFLD coordinates of message
30800 COLOR 7,0:LOCATE YFLD,XFLD,0:PRINT"Press any key.....";:GOS
UB 30030
30810 IF SND%>1 THEN BEEP
30820 C$=INKEY$:IF C$="" THEN 30820
30830 IF C$=CHR$(27) THEN ESCK=1 ELSE ESCK=0
30840 RETURN
30850 COLOR 7,0:PRINT CHR$(13) CHR$(13) SPC(29)"Press any key.....
.";:GOSUB 30030:GOTO 30810
30860 '
30870 'INPUT/PRINT MATRIX ROUTINE
30880 'Caller specifies MAT, XFLD, YFLD, NUMDIG, P=1 PRINT
30890 COLOR 12,0:LOCATE YFLD,XFLD,0:PRINT COM$:YFLD=YFLD+2:XFLD=1:I
NFLG=3
30900 FOR I=1 TO ROW%:XOFF=6:NC=0:COLOR 12,0:LOCATE YFLD,XFLD:PRINT
"R";
30910 LOCATE YFLD,XFLD+1:PRINT I;:COLOR 6,0:XFLD=6:FOR J=1 TO COL%:
NC=NC+1
30920 IF P=1 THEN 30980 ELSE GOSUB 7110:IF ESCK=1 THEN RETURN
30930 IF MAT=1 THEN A(I,J)=S:GOTO 31040
30980 LOCATE YFLD,XFLD,0
30990 IF MAT=1 THEN PRINT A(I,J);:GOTO 31040
31010 IF MAT=3 THEN PRINT C(I,J);:GOTO 31040
31035 IF MAT=6 THEN PRINT SPO(I,J);
31040 XFLD=NC*(NUMDIG+2)+XOFF:IF XFLD>70 THEN YFLD=YFLD+1:XFLD=10:X
OFF=10:NC=0:IF YFLD>22 THEN GOSUB 31070
31050 NEXT J:YFLD=YFLD+1:IF YFLD>22 THEN GOSUB 31070

```

```

31060 XFLD=1:NEXT I:RETURN
31070 SXFLD=XFLD:XFLD=28:YFLD=24:GOSUB 30800:YFLD=3:XFLD=SXFLD:COLO
R 6,0:CLS:RETURN
31080 '
31090 'READ MATRIX FROM DISK FILE
31100 'Caller specifies MAT, FILENUM, ROW%, COL%, & opens file
31110 ON ERROR GOTO 31220
31120 FOR I=1 TO ROW%:FOR J=1 TO COL%:IF EOF(FILENUM) THEN 31210
31130 IF MAT=0 THEN INPUT #FILENUM,KDIM(I):GOTO 31200
31140 IF MAT=1 THEN INPUT #FILENUM,A(I,J):GOTO 31200
31200 NEXT J,I:FERR$='OK':RETURN
31210 FERR$='END OF FILE BEFORE FINISHED':RETURN
31220 ON ERROR GOTO 0:GOSUB 31260:RESUME 31230
31230 RETURN
31240 '
31250 'ERROR CHECKING ROUTINE
31260 IF ERR=64 THEN FERR$='BAD FILE NAME':RETURN
31270 IF ERR=70 THEN FERR$='DISK WRITE PROTECT':RETURN
31280 IF ERR=53 THEN FERR$='FILE NOT FOUND':RETURN
31290 IF ERR=67 THEN FERR$='TOO MANY FILES ON DISK':RETURN
31300 IF ERR=55 THEN FERR$='FILE ALREADY IN USE':RETURN
31310 IF ERR=71 THEN FERR$='DISK NOT READY':RETURN
31320 IF ERR=61 THEN FERR$='DISK FULL':RETURN
31330 IF ERR=72 THEN FERR$='BAD DISK MEDIA':RETURN
31340 FERR$='DISK I/O ERROR':RETURN
31350 '
31360 'SAVE MATRIX ON DISK
31370 'Caller specifies MAT, FILENUM, ROW%, COL% & opens file
31380 ON ERROR GOTO 31480
31390 FOR I=1 TO ROW%:FOR J=1 TO COL%
31400 IF MAT=0 THEN PRINT #FILENUM,KDIM(I);:GOTO 31470
31410 IF MAT=1 THEN PRINT #FILENUM,A(I,J);:GOTO 31470
31470 NEXT J,I:FERR$='OK':ON ERROR GOTO 0:RETURN
31480 ON ERROR GOTO 0:GOSUB 31260:RESUME 31490
31490 RETURN
31500 '
31510 'KILL FILE ROUTINE
31520 'Caller specifies FILENAM$
31530 ON ERROR GOTO 31550:KILL FILENAM$:FERR$="OK"
31540 ON ERROR GOTO 0:RETURN
31550 GOSUB 31260:RESUME 31540
31570 '
31580 'CHANGE MATRIX ELEMENT
31590 'Caller specifies MAT, ROW1%, COL1%, & S (the new value)
31600 IF MAT=0 THEN KDIM(ROW1%)=S:RETURN
31610 IF MAT=1 THEN A(ROW1%,COL1%)=S:RETURN
31660 '
31670 'ERASE ANY LINE ROUTINE
31680 LOCATE YFLD,1,0:PRINT SPC(79);:RETURN
31690 '
31700 'WRITE BLANK TO ARCHIVE

```

```

31710 COM$="":GOSUB 30410:RETURN
31720 '
31730 'STATE TRANSITION MATRIX CALCULATION
31740 E=.001:L%=50:FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1)
31750 IF I=J THEN SDI(I,J)=1 ELSE SDI(I,J)=0
31760 SP1(I,J)=T*SDI(I,J):SG(I,J)=T*A(I,J):NEXT J,I:N%=1
31770 N%=N%+1:FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1)
31780 SW(I,J)=T/N%*SG(I,J):SP0(I,J)=SP1(I,J):SP1(I,J)=SP1(I,J)+SW(I,J)
31790 SG(I,J)=0:NEXT J,I
31800 FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1):FOR K%=1 TO KDIM(1)
31810 SG(I,J)=SG(I,J)+SW(I,K%)*A(K%,J):NEXT K%,J,I
31820 IF N%<L% THEN 31860 ELSE GOSUB 30760:CLS:COLOR 15,0
31830 COM$="STATE TRANSITION MATRIX NOT CONVERGING PROPERLY!":IF AR
C%=1 THEN GOSUB 30410
31840 LOCATE 10,15,0:PRINT COM$:COLOR 7,0:LOCATE 12,13,0:PRINT"Plea
se re-evaluate model or try smaller time step"
31850 XFLD=28:YFLD=14:GOSUB 30800:RS=0:RETURN
31860 FOR Q%=1 TO KDIM(1):FOR R%=1 TO KDIM(1)
31870 IF SP1(Q%,R%)=0 THEN 31890
31880 IF E<ABS((SP1(Q%,R%)-SP0(Q%,R%))/SP1(Q%,R%)) THEN 31770
31890 NEXT R%,Q%
31900 FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1):SG(I,J)=0:FOR K%=1 TO K
DIM(1)
31910 SG(I,J)=SG(I,J)+A(I,K%)*SP1(K%,J):NEXT K%,J,I
31920 FOR I=1 TO KDIM(1):FOR J=1 TO KDIM(1):SP0(I,J)=SDI(I,J)+SG(I,J)
31930 NEXT J,I:RETURN
31933 '
32300 'NO MODEL/MATRIX ERROR ROUTINE
32310 GOSUB 30760:CLS:COLOR 15,0:LOCATE 10,28,0:PRINT COM$:XFLD=28:
YFLD=12:GOSUB 30800:RETURN
32320 '
32330 'ERASE EXISTING ERROR ROUTINE
32340 GOSUB 30760:CLS:COLOR 15,0:LOCATE 10,24,0:PRINT COM$:COLOR 7,
0:INFLG=1:GOSUB 7120:RETURN
32350 '
32430 'MODIFY ANY MATRIX ROUTINE
32440 CLS:XFLD=1:YFLD=3:NUMDIG=10:P=1:GOSUB 30890:COLOR 12,0
32450 LOCATE 25,1,0:PRINT"<ENTER> re-input matrix, ESC quit, else R
OW >";
32460 LOCATE 25,46,0:PRINT SPC(33):XFLD=48:YFLD=25:NUMDIG=10:INFLG=
3
32470 GOSUB 7110:IF ESCK=1 THEN RETURN
32480 IF S<>0 THEN 32510 ELSE XFLD=1:YFLD=3:CLS:P=0:GOSUB 30890
32490 IF ESCK<1 THEN RETURN ELSE CLS:COLOR 15,0:LOCATE 10,27,0
32500 PRINT"WARNING! MATRIX INCOMPLETE":XFLD=30:YFLD=12:GOSUB 3076
0:GOSUB 30800:RETURN
32510 IF S>ROW% OR S<1 THEN GOSUB 30760:GOTO 32460
32520 ROW1%=S:LOCATE 25,46,0:PRINT ROW1% SPC(5);:LOCATE 25,50,0:COL
OR 12,0:PRINT"COL >";

```

```

32530 LOCATE 25,55,0:PRINT SPC(23):XFLD=56:YFLD=25:GOSUB 7110:IF ES
CK=1 THEN RETURN
32540 IF S>COL% OR S<1 THEN GOSUB 30760:GOTO 32530
32550 COL1%-S:LOCATE 25,55,0:PRINT COL1% SPC(6);:LOCATE 25,59,0:COL
OR 12,0:PRINT"VALUE >";:XFLD=67:YFLD=25
32560 GOSUB 7110:IF ESCK=1 THEN RETURN ELSE GOSUB 31600:GOTO 32440
32590 '
32600 'MATRIX CALCULATIONS MENU
32610 CLS:WIDTH 80:COLOR 12,0:LOCATE 6,30,0:PRINT"MATRIX CALCULATIO
NS"
32620 COLOR 3,0:LOCATE 9,23,0:PRINT"<Sft-F1>....DETERMINANT"
32630 LOCATE 10,23,0:PRINT"<Sft-F2>....INVERSE"
32640 LOCATE 11,23,0:PRINT"<Sft-F3>....CHARACTERISTIC POLYNOMIAL"
32650 LOCATE 12,23,0:PRINT"<Sft-F4>....EIGENVALUES"
32660 LOCATE 13,23,0:PRINT"<Sft-F5>....STATE TRANSITION MATRIX"
32670 LOCATE 14,23,0:PRINT"<Sft-F6>....RESOLVENT"
32680 LOCATE 15,23,0:PRINT"<ESC>.....EXIT TO MATRIX MENU"
32690 COLOR 12,0:LOCATE 18,25,0:PRINT"PRESS DESIRED FUNCTION KEY":R
ETURN
32695 '
32700 GOSUB 30760:SCREEN 0,1:WIDTH 80:CLS:COLOR 15,0:LOCATE 10,28,0
:PRINT COM$
32710 COLOR 7,0:LOCATE 12,33,0:PRINT RT$:INFLG=1:GOSUB 7120:RETURN
33000 IF N=1 THEN SPO(N,1)=-D(N,1):SPO(N,2)=0:RETURN
33010 IF N=3 THEN 33070
33020 IF N=4 THEN 33200
33030 SR=D(2,1)*2-4*D(1,1):SI=0:N=2:GOSUB 33600
33040 IF RI<>0 THEN 33060 ELSE SPO(1,2)=0:SPO(2,2)=0
33050 SPO(1,1)=(-D(2,1)+RR)/2:SPO(2,1)=(-D(2,1)-RR)/2:RETURN
33060 SPO(1,1)=-D(2,1)/2:SPO(2,1)=SPO(1,1):SPO(1,2)=RI/2:SPO(2,2)=-
RI/2:RETURN
33070 A=(3*D(2,1)-D(3,1)*2)/3:B=(2*D(3,1)*3-9*D(3,1)*D(2,1)+27*D(1,
1))/27
33080 SR=B*2/4+A*3/27:SI=0:N=2:GOSUB 33600
33090 QTR=RR:QTI=RI:SR=QTR-B/2:SI=QTI:N=3
33100 IF SR<0 AND SI=0 THEN SR=-SR:GOSUB 33600:RR=-RR:GOTO 33120
33110 GOSUB 33600
33120 QAR=RR:QAI=RI:SR=-QTR-B/2:SI=-QTI
33130 IF SR<0 AND SI=0 THEN SR=-SR:GOSUB 33600:RR=-RR:GOTO 33150
33140 GOSUB 33600
33150 QBR=RR:QBI=RI:SR=-3:SI=0:N=2:GOSUB 33600
33160 SPO(1,1)=QAR+QBR:SPO(1,2)=QAI+QBI:QTR=(QAR-QBR)/2:QTI=(QAI-QB
I)/2
33170 SR=FNCRM(QTR,QTI,RR,RI):SI=FNCIM(QTR,QTI,RR,RI)
33180 SSR=-(QAR+QBR)/2:SSI=-(QAI+QBI)/2:SPO(2,1)=SSR+SR:SPO(2,2)=SS
I+SI:SPO(3,1)=SSR-SR:SPO(3,2)=SSI-SI
33190 FOR I=1 TO 3:SPO(I,1)=SPO(I,1)-D(3,1)/3:NEXT I:RETURN
33200 A=D(4,1):B=D(3,1):C=D(2,1):D=D(1,1)
33210 FOR I=1 TO 4:SPO(I,1)=0:SPO(I,2)=0:NEXT I:PI=3.1415926#:PIO2=
1.5707963#
33220 QP=-B:QQ=A*C-4*D:QR=-(A*2)*D+4*B*D-C*2

```

```

33230 QCA=(3*Q-QP^2)/3: QCB=(2*QP^3-9*QP*Q+27*QR)/27
33240 QTEMP=QCB^2/4+QCA^3/27: N=2: SR=QTEMP: SI=0
33250 GOSUB 33600: QTR=RR: QTI=RI
33260 SR=QTR-QCB/2: SI=QTI: N=3
33270 IF SR<0 AND SI=0 THEN SR=-SR: GOSUB 33600: RR=-RR: GOTO 33290
33280 GOSUB 33600
33290 QCAAR=RR: QCAAI=RI: SR=-QTR-QCB/2: SI=-QTI
33300 IF SR<0 AND SI=0 THEN SR=-SR: GOSUB 33600: RR=-RR: GOTO 33320
33310 GOSUB 33600
33320 YR=QCAAR+RR-QP/3: YI=QCAAI+RI
33330 SR=A^2/4-B+YR: SI=YI: N=2: GOSUB 33600
33340 QRR=RR: QRI=RI
33350 IF QRR=0 AND QRI=0 THEN 33450
33360 QTEMP=4*A*B-8*C-A^3
33370 QTR=FNCRD(QTEMP,0,4*QRR,4*QRI)
33380 QTI=FNCID(QTEMP,0,4*QRR,4*QRI)
33390 QRTR=FNCRM(QRR,QRI,QRR,QRI)
33400 QRTI=FNCIM(QRR,QRI,QRR,QRI)
33410 SR=3*A^2/4-QRTR-2*B+QTR: N=2: SI=-QRTI+QTI: GOSUB 33600
33420 QDR=RR: QDI=RI
33430 SR=3*A^2/4-QRTR-2*B-QTR: N=2: SI=-QRTI-QTI: GOSUB 33600
33440 QER=RR: QEI=RI: GOTO 33500
33450 QTR=FNCRM(YR,YI,YR,YI)-4*D
33460 QTI=FNCIM(YR,YI,YR,YI)
33470 SR=QTR: SI=QTI: N=2: GOSUB 33600: QTR=2*RR: QTI=2*RI
33480 SR=QTR+3*A^2/4-2*B: SI=QTI: GOSUB 33600: QDR=RR: QDI=RI
33490 SR=-QTR+3*A^2/4-2*B: SI=-QTI: GOSUB 33600: QER=RR: QEI=RI
33500 SP0(1,1)=-A/4+QRR/2+QDR/2
33510 SP0(1,2)=QRI/2+QDI/2
33520 SP0(2,1)=-A/4+QRR/2-QDR/2
33530 SP0(2,2)=QRI/2-QDI/2
33540 SP0(3,1)=-A/4-QRR/2+QER/2
33550 SP0(3,2)=-QRI/2+QEI/2
33560 SP0(4,1)=-A/4-QRR/2-QER/2
33570 SP0(4,2)=-QRI/2-QEI/2
33580 RETURN
33590 'This subroutine finds the Nth root of a complex number.
33600 PI=3.1415926#: PIO2=1.5707963#
33605 IF SI=0 AND SR=0 THEN RI=0: RR=0: RETURN
33610 IF SI<0 AND SR<0 THEN ANGLE=-PI+ATN(SI/SR): GOTO 33690
33620 IF SI<0 AND SR=0 THEN ANGLE=-PIO2: GOTO 33690
33630 IF SI<0 AND SR>0 THEN ANGLE=-ATN(-SI/SR): GOTO 33690
33640 IF SI=0 AND SR<0 THEN ANGLE=PI: GOTO 33690
33650 IF SI=0 AND SR>0 THEN ANGLE=0: GOTO 33690
33660 IF SI>0 AND SR<0 THEN ANGLE=PI-ATN(-SI/SR): GOTO 33690
33670 IF SI>0 AND SR=0 THEN ANGLE=PIO2: GOTO 33690
33680 IF SI>0 AND SR>0 THEN ANGLE=ATN(SI/SR)
33690 NR=SQR(SI^2+SR^2): RR=NR*(1/N)*COS(ANGLE/N): RI=NR*(1/N)*SIN(ANGLE/N)
33700 RETURN

```

**The vita has been removed from
the scanned document**

DACS: AN INTERACTIVE COMPUTER PROGRAM TO AID IN THE DESIGN
AND ANALYSIS OF LINEAR CONTROL SYSTEMS

by

W. Ivan Lewis, III

(ABSTRACT)

DACS is an interactive computer program for the IBM-PC that aids in the design and analysis of linear control systems. Written in compiled basic, DACS includes Root Locus, Bode plots, Nyquist diagrams, Nichols charts, system simulation, and calculates the system time response in closed-form. The state-space description is used in the simulation and time response segments while the system transfer function is used in the root locus and frequency response segments. Calculated data may be displayed in either graphical or tabular form. Graphics features of DACS include automatic-scaling, zooming, graph coordinates derived from cursor location, graphics screen dumps to disk or printer, and redisplaying of disk files. DACS is menu driven and the majority of input/output is accomplished through the function keys. All system models and data may be saved on and recalled from disk. Help screens, three levels of sound, color, and session archiving are also provided. DACS provides a wide variety of linear control system analysis tools for the engineering desktop.