

AD-A166 479

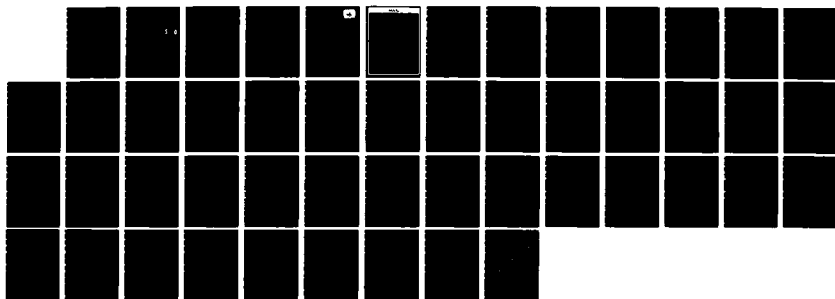
CONFIDENCE ELLIPSE RESEARCH SOFTWARE (EAFF)(U) JET
PROPULSION LAB PASADENA CA J MYHRE ET AL 08 AUG 85
JPL-D-2786 NAS7-918

1/1

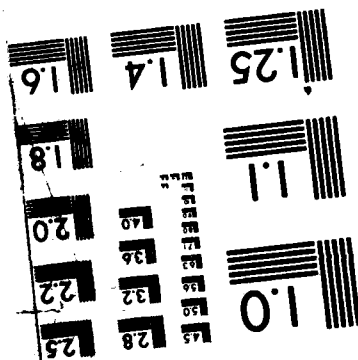
UNCLASSIFIED

F/G 9/2

NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



U.S. ARMY INTELLIGENCE CENTER AND SCHOOL
Software Analysis and Management System

Confidence Ellipse Research Software

EAAF

Technical Memorandum No. 6

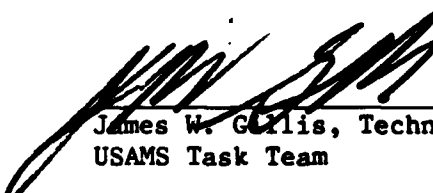
August 8, 1985

by

Dr. Janet Myhre
Mr. Will Duquette
Mr. Daniel Hockman

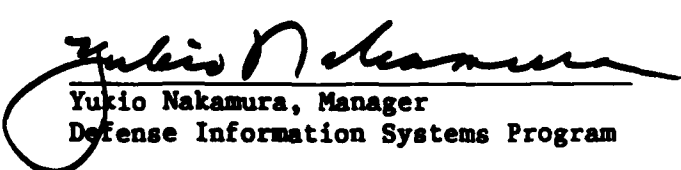
DTIC
ELECTE
APR 09 1986
S D

Concur:


James W. Gillis, Technical Staff
USAMS Task Team


Edward J. Records, Manager
USAMS Task Team


J. P. McClure, Manager
Ground Data Systems Section


Yukio Nakamura, Manager
Defense Information Systems Program

JET PROPULSION LABORATORY
California Institute of Technology
Pasadena, California

JPL D-2786

DTIC FILE COPY

86 4 8 069

AD-A166 479

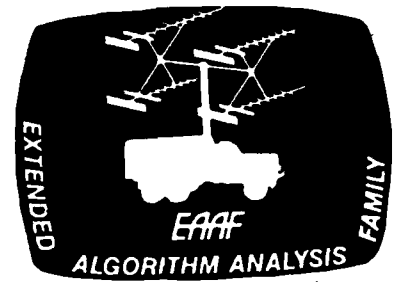
REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER JPL D-2786	2. GOVT ACCESSION NO. AD-A166479	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CONFIDENCE ELLIPSE RESEARCH SOFTWARE		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER No. 41
7. AUTHOR(s) Dr. Janet Myhre, Will Duquette, Daniel Hockman		8. CONTRACT OR GRANT NUMBER(s) NAS7-918
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematical Analysis Research Corp. 4239 Via Padova Claremont, CA 91711		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS RE 182 AMEND # 187
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, USAICS ATTN: ATSI-CD-SF Ft. Huachuca, AZ 85613-7000		12. REPORT DATE 8 August, 1985
		13. NUMBER OF PAGES 42
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Jet Propulsion Laboratory California Institute of Technology 4800 Oak Grove Drive, ATTN: 126-200 Pasadena, CA 91109		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Distribution		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Prepared under contract to JPL for the U. S. Army Intelligence Center and School's Combat Developer's Support Facility		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) CONFIDENCE ELLIPSE, ELLIPTICAL ERROR PROBABLE (EEP), COVARIANCE MATRIX, BASIC, STATISTICAL TESTS, MATHEMATICAL SIMULATION		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This is one of a series of algorithm analysis reports performed for the US Army Intelligence Center and School covering selected algorithms in existing or planned Intelligence and Electronic Warfare (IEW) systems. This report documents the software used in the analysis of ellipse combination and testing which was reported in report 40 of this series.		

PREFACE

The work described in this publication was sponsored by the United States Army Intelligence Center and School. The writing and publication of this paper was supported by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

UNCLASSIFIED



CONFIDENCE ELLIPSE RESEARCH SOFTWARE

EAAF

Technical Memorandum No. 6

August 8, 1985

by

Dr. Janet Myhre
Mr. Will Duquette
Mr. Daniel Hockman

JET PROPULSION LABORATORY
California Institute of Technology
Pasadena, California

UNCLASSIFIED



MATHEMATICAL ANALYSIS RESEARCH CORPORATION

NO. 41

CONFIDENCE ELLIPSE RESEARCH SOFTWARE

8 AUGUST 1985

4239 VIA PADOVA, CLAREMONT, CA 91711

TABLE OF CONTENTS

- I. INTRODUCTION
- II. GUIDE TO GENELLIPSE:
 Ellipse Graphing Program
- III. GUIDE TO ELLSIM:
 Ellipse Simulation Program
- IV. GUIDE TO ELLIPSTUFF:
 Ellipse Routine Library

PROGRAM LISTINGS

- A. GENELLIPSE
- B. ELLSIM
- C. ELLIPSTUFF

I. INTRODUCTION

three
This document describes ~~several~~ programs used by MARC in studying and testing confidence ellipses. All of these programs were developed on a Hewlett Packard Series 200 computer, the HP9836, and are written in Hewlett Packard's BASIC 3.0 programming language. This a very powerful version of BASIC, and consequently many of the programming constructs found in these programs are non-standard. Also, the graphics commands are peculiar to this BASIC.

The programs are called Genellipse, Ellsim, and Ellipstuff. The listings and comments are current as of August 9, 1985. The programs themselves are subject to change without notice.

Genellipse is an "ellipse graphing and combination" program. It allows the user to specify up to twenty confidence ellipses, combine ellipses, graph ellipses, and test ellipses for combination. It has been used primarily to explore the geometric properties of the combination method, but has also been used to create figures for a number of other reports.

Ellsim is a "confidence ellipse simulation" program. It has been used to explore the robustness and properties of the statistical test (which is used to decide whether or not to combine two ellipses). In it, two normal data distributions are specified, corresponding to two emitters. Confidence ellipses are generated, tested, and combined, and various descriptive statistics are compiled. In addition, there is a routine to calculate the power of the statistical test in certain cases.

Ellipstuff is an "ellipse routine library," which contains a large number of routines to make working with ellipses easier. As such, both Genellipse and Ellsim contain all of these routines, but they are also listed separately. If more programs need to be written, these are the routines to build them with.

Each of these programs is listed below, along with a short description of their use. This is a supplement to the report "Testing and Combination of Confidence Ellipses: A Geometric Analysis," submitted to JPL by MARC on August 5, 1985. Refer here any questions concerning confidence ellipses, the combination method, or the statistical test used.

II. GUIDE TO GENELLIPSE: Ellipse Graphing Program

Genellipse stands for General Ellipse graphing, and can also combine ellipses and test two ellipses for combination. The program makes extensive use of the Ellipstuff Library Routines.

When Genellipse is executed, the following menu is displayed on the screen:

GENERAL ELLIPSE GRAPHER

- (A) -- Enter Ellipse Mean Point (Center)
- (B) -- Enter Ellipse Shape (Covariance Matrix)
- (C) -- Enter Ellipse Shape (Axes and Orientation)
- (D) -- Combine Two Ellipse (JPL Method)
- (E) -- Graph an Ellipse
- (F) -- Clear Graphics Screen
- (G) -- Choose Plotter
- (H) -- Set Graphics Screen Bounds
- (I) -- Calculate Acceptance Test Statistic (Chi-square)
- (J) -- Draw Axes
- (X) -- Exit Program

Enter your choice:

These options will be dealt with one by one.

(A) -- Enter Ellipse Mean Point (Center)

This option prompts the user to enter the center point for one or more ellipses. The program first asks for the ellipse (1-20), and then for the mean point. This will repeat until all means have been entered. When the user is through entering means, pressing the return key in response to the question "Getting mean point for Ellipse #" will return the user to the menu shown above.

(B) -- Enter Ellipse Shape (Covariance Matrix)

There are two ways of specifying the shape of a confidence ellipse. The first is through a covariance matrix. This option will ask for the ellipse number, and then prompt for the elements of the matrix. It repeats until all matrices have been entered, just as option A does.

(C) -- Enter Ellipse Shape (Axes and Orientation)

The other way of specifying the shape of a confidence ellipse is geometrically. This option prompts for the ellipse number, just as those above, and then asks for the axes lengths and orientation in degrees of the ellipse. It repeats until all ellipses have been entered, just as option A does.

(D) -- Combine Two Ellipses (JPL Method)

This option combines confidence ellipses using the method described in "Testing and Combination of Confidence Ellipses: A Geometric Analysis." The option prompts for the numbers of the two ellipses to be combined, and then for the number of the resultant ellipse. For example, suppose the user had entered ellipses 1, 2, 3, and 7 out of 20, and wanted to combine 2 and 3. The resultant ellipse could be given number 4, 5, 6, or 8 through 20. Further, if either of ellipses 1 or 7 were no longer needed, the resultant ellipse could be given either of numbers 1 or 7 as well. This would, of course, erase the ellipses originally stored in these slots.

(E) -- Graph an Ellipse

This option prompts for the number of the ellipse the user wishes to graph, and continues asking for ellipses until all desired have been graphed. The process is similar to that in option A. The ellipse will be graphed on the current graphics device (See option G). Other options related to graphing are G, H, and J.

(F) -- Clear Graphics Screen

If the CRT screen is the current graphics device, choosing this option will erase all ellipses currently drawn on it.

(G) -- Choose Plotter

This options allows the user to select the current graphics device. This program is currently written to graph on the CRT screen and on an HP7470A two-pen plotter. Consequently, if selected, the program will ask "Do you want to (D)raw, (E)rase, Plot with Pen (0), (1), (2)?" Choosing "D" will cause it to do all graphing on the screen. Choosing "E" will cause it to graph on the screen in "black" -- so that an ellipse may be erased without clearing the whole screen. Choosing (1) or (2) will cause it to graph on the plotter, using the specified pen. Choosing (0) causes the plotter to put away the pen that its using, and then sets the CRT screen to be the graphics device.

(H) -- Set Graphics Screen Bounds

This option is used to scale the graphics screen. Default scaling is -50 to 50 on the X-axis and -50 to 50 on the Y-axis. When this option is chosen, it will first present the "limits" of any ellipse: that is, how far the ellipse extends in the X and Y directions, so that a reasonable screen size may be chosen. The program prompts for the number of each

ellipse, just as in option A. Press the return key after all desired ellipse limits have been seen. The program will then ask for the screen bounds: Minimum X value, Maximum X value, Minimum Y value, Maximum Y value.

(I) -- Calculate Acceptance Test Statistic (Chi-square)

This option prompts the user for two ellipse numbers, and then performs the statistical test described in "Testing and Combining Confidence Ellipses: A Geometrical Analysis." The value of the test statistic is printed on the screen. If it is less than or equal to the 95% Chi-square value, 5.991, then the test accepts; otherwise the test rejects. (Actually, the test may be run at any confidence level desired. However, the ellipses used in this program are assumed to be 95% confidence ellipses, and if the test is performed at any other level the geometrical results shown in "Testing and Combining..." concerning the test will not necessarily hold true.

(J) -- Draw Axes

This option draws a set of axes on the current graphics device, and labels them according to the current screen bounds (see Options G and H).

III. GUIDE TO ELLSIM: Ellipse Simulation Program

Ellsim stands for Ellipse Simulator, but can also find analytical estimates of statistical power. It uses the Ellipstuff Library extensively.

See the report "Testing and Combining Confidence Ellipses: A Geometric Analysis" for a description of simulation and how it has been applied to the ellipse combination problem. This is the program used to generate the results in section V of that report. Note that in this program, sensor error is assumed to follow a bivariate normal distribution about the true location of the emitter. When location estimates are derived from lines of bearing, however, this assumption may be unrealistic.

When the program is run, it will present the user with the following menu of choices:

Ellipse Combination Program Driver

- (A) -- Specify True Covariance Matrices
- (B) -- Specify True Mean Parameters
- (C) -- Specify Observations
- (D) -- Call Simulation Generator
- (E) -- Call Power Generator
- (X) -- Exit Program

Enter your choice:

Options A, B, and C are used to specify the two data distributions.

Option A prompts for the means of two bivariate normal distributions. Setting the means to be equal is equivalent to having only one emitter. Setting them apart is equivalent to having two emitters.

Option B prompts for the covariance matrices for the two bivariate normal distributions.

Option C prompts for the sample sizes to be used for each distribution. Note that any confidence ellipses generated will have covariance matrices equal to those specified in option B, divided by these sample sizes.

Options D and E call the Simulation Generator and the Power Generator respectively. These will be discussed individually.

(D) -- The Simulation Generator

Choosing this option leads to another menu: the simulation menu. It appears as follows:

Enter the Letter of your choice:

- (A) Reset the Random Number Seed
- (B) Simulate using true Covariance Matrices
- (C) Simulate using estimated Covariance Matrices
- (D) Display Results on the Printer
- (X) Exit Program

Enter Your Choice:

These options will be dealt with one by one

(A) -- Reset the Random Number Seed

This option allows the user to start the random number process with a given seed; this is generally not necessary, but is useful for purposes of debugging.

(B) -- Simulate using true Covariance Matrices

This option will prompt the user for the number of simulations (generally 100 or more), and then proceeds in this manner: for each simulation it generates the number of observations specified by the sample size entered earlier. It estimates the emitter locations from these observations, and calculates confidence ellipses. It tests to see if these ellipses may be combined. Finally, it checks to see if the combined ellipse contains the true location of the emitter(s). When all simulations are done, it compiles these results. See "Testing and Combining Confidence Ellipses" for more information.

(C) -- Simulate using estimated Covariance Matrices

The process here is the same as that outlined for option B, except for one addition: the covariance matrices used in the confidence ellipses and statistical test are estimated. However, the formulas for the ellipses and test assume that the covariance matrices are known. Thus, this option is used to explore what happens if estimated matrices are mistakenly used. Note that the estimates are made using the " S^2 " statistic. This is the usual way of estimating variance-covariance from a data set, but is different from the methods used in most if not all of the position fixing algorithms we have seen.

(D) -- Display Results on the Printer

When either option B or option C has been completed, the results of the simulations are shown in the screen. If a hardcopy is desired, selecting this option will cause the results of the last simulation run to be output on the printer.

(X) -- Exit Program

This option will exit the Simulation Generator, and return to the original menu.

(E) -- The Power Generator

The power of a statistical test is essentially the probability that the test will reject when it ought to. That is, it is the probability that the statistical test will say that there are two emitters when in fact there are two emitters. In the problem at hand, however, the power is not a single quantity; in fact, there is a different power value for each pair of emitters. If the emitters are close together, the power of the test will be low; if they are far apart, the power will be close to 1.

This option works in the following way, for convenience sake. It uses the covariance matrices and sample sizes specified from the main menu, but allows the user to enter the distance between the two emitters. It calculates the power, and then asks for another distance. To return to the main menu, enter 0 for the distance.

IV. GUIDE TO ELLIPSTUFF: Ellipse Routine Library

Ellipstuff is a library of subprograms and functions written to facilitate confidence ellipse research for JPL. It covers such things as defining ellipses, combining ellipses, testing ellipses for combination, and graphing ellipses.

To use Ellipstuff, simply include it in within a program. Genellipse and Ellsim are examples of this.

Confidence ellipses and how they are stored

A confidence ellipse is defined by two things: a point estimate, or mean, and a covariance matrix. Thus, both of these pieces of information must be stored for each ellipse. In addition, it is often necessary to have the inverse of the covariance matrix on hand as well. Ellipstuff stores ellipses in a matrix with 40 slots, allowing the storage of 20 covariance matrices with their inverses. In general most of the Ellipstuff routines deal with the inverses themselves, but in case it is necessary to use them explicitly, the ellipses are stored in slots 1 through 20 and the inverse covariances are stored in slots 21 through 40. By convention, the inverse of the covariance matrix for the ellipse in, say, slot 3, is stored in slot 23, and so on. Examine the routines for more programming information.

Ellipstuff User Routines

The slots specified in the following routines should be between 1 and 20 inclusive.

Get_ell_mean(Ellipse)

This command prompts the user to input the mean (center point) for the ellipse in slot Ellipse.

Example: Get_ell_mean(1)

Get_covariance(Ellipse)

This command prompts the user to input the covariance matrix for the ellipse in slot Ellipse.

Example: Get_covariance(2)

Get_axes(Ellipse)

This command prompts the user to input the shape of the ellipse (lengths of the semi-minor and semi-major axes, and the orientation) which is then converted to a covariance matrix.

Example: Get_axes(3)

Test(Ellipse1, Ellipse2, Work, Test_stat)

This command runs a chi-square test on the ellipses in slots Ellipse1 and Ellipse2 in order to see if they may be combined. The value of the test-statistic is returned in Test_stat. Work is the number of any unused slot, to be used for scratch work.

Example: Test(3,4,20,Some_variable)

Combine_ellipse(Ellipse1, Ellipse2, Combo)

This command combines the ellipses and point estimates in slots Ellipse1 and Ellipse2, and stores the combined ellipse in slot Combo.

Example: Combine_ellipse(3,4,5)

Draw_ellipse(Ellipse, Xmin, Xmax, Ymin, Ymax, Prob_constant)

This draws the ellipse in slot Ellipse on the current plotting device (see Choose_plotter). Xmin, Xmax, Ymin, and Ymax specify the screen dimensions. The default values are (-50,50,-50,50).

Prob_constant is zero minus the chi-square cutoff associated with the confidence level. (The cutoff is $P = -2\text{LN}(1-\text{Alpha})$, where Alpha is the confidence level. For some reason, all of the programming was done in terms of $-P = 2\text{LN}(1-\text{Alpha})$. Thus, for 95% ellipses, Prob_constant = -5.991).

Example: Draw_ellipse(2,-10,10,-20,20,-5.991)

Disp_extremes(Ellipse)

This command displays the extreme x and y points calculated by Get_bound for the ellipse in slot Ellipse. This is designed to aid in choosing the Xmin, Xmax, Ymin, and Ymax values required by Draw_ellipse.

Example: Disp_extremes(1)

Display_cov(Ellipse)

This command displays the covariance matrix of the ellipse in slot Ellipse on the crt screen (not graphically). It may also be used to display the inverses in slots 21 through 40.

Example: Display_cov(11)

Choose_plotter

When first initialized, Ellipstuff assumes that all graphing will be done on the CRT. Choose_plotter is called to allow the program user to select which device to use. Note: this program was written on a system with two graphics devices--a CRT screen and an HP7470A two pen plotter. Choose_plotter presents the user with 5 options:

D)raw -- graph on the CRT in the normal fashion (DEFAULT).

E)rase -- graph on the CRT in "black". This may be used to erase things.

1) -- graph on the plotter using Pen 1.

2) -- graph on the plotter using Pen 2.

0) -- put away the current plotter pen and select the CRT.

Example: Choose_plotter

Ellipstuff Low Level Routines

These are routines which are used in building user routines. Since these must deal with inverse covariance matrices also, the slots specified may run from 1 to 40. The user will generally not use these, except when adding to the Ellipstuff module.

Invert(Source, Destination)

This command inverts the covariance matrix in slot Source and puts the inverse in slot Destination.

Example: Invert(3, 23)

Add_covariance(Ellipse1, Ellipse2, Summer)

This command adds the covariance matrices stored in slots Ellipse1 and Ellipse2, and stores them in slot Summer.

Example: Add_covariance(1,2,3)

Get_bounds(Ellipse, Prob_constant)

This command calculates and saves the extreme x and y points for the ellipse in slot Ellipse.

Example: Get_bounds(3, -5.991)

A. GENELLIPSE: Ellipse Graphing Program

Genellipse is an "ellipse graphing and combination" program. It allows the user to specify up to twenty confidence ellipses, combine ellipses, graph ellipses, and test ellipses for combination. It has been used primarily to explore the geometric properties of the combination method, but has also been used to create figures for a number of other reports.

```

100  !*****!
110  !*          GENERAL ELLIPSE GRAPHER          *!
120  !*          *!
130  !* This program uses the general ellipse generation and graph-*!
140  !* ing routines found in the file ELLIPSTUFF. Its purpose *!
150  !* is to graph general ellipses and combinations of ellipses. *!
160  !*          *!
170  !*          Will Duquette          May 13, 1985          *!
180  !*****!
190  COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(
20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
191  REAL Xmin,Xmax,Ymin,Ymax,Prob_constant
200  Init prog:
210  GINIT
220  GCLEAR
230  ! These variables define the size of the screen. For best results,
240  ! the lengths of the X and Y axes should be about the same.
250  Xmin=-50
260  Xmax=50
270  Ymin=-50
280  Ymax=50
281  Prob_constant=2*LOG(.05)
290  ! Display the menu
300  Menu top:
310  Clearscreen
320  PRINT "GENERAL ELLIPSE GRAPHER"
330  PRINT
340  PRINT "-----"
350  PRINT " (A) -- Enter Ellipse Mean Point (Center)"
360  PRINT " (B) -- Enter Ellipse Shape (Covariance Matrix)"
370  PRINT " (C) -- Enter Ellipse Shape (Axes and Orientation)"
380  PRINT " (D) -- Combine Two Ellipses (JPL Method)"
390  PRINT " (E) -- Graph an Ellipse"
400  PRINT " (F) -- Clear Graphics Screen"
410  PRINT " (G) -- Choose Plotter"
420  PRINT " (H) -- Set Graphics Screen Bounds"
430  PRINT " (I) -- Calculate Acceptance Test Statistic (Chi-square)"
440  PRINT " (J) -- Draw Axes"
450  PRINT " (K) -- Exit Program"
460  PRINT
470  PRINT "Enter your choice:"
480  DISP "Choose an option:"
490  INPUT Option$
500  SELECT Option$
510  CASE "A","a" ! Get ellipse mean points.
520  Clearscreen
530  PRINT "Getting mean point for Ellipse#";
540  REPEAT
550  Ellipse=0
560  DISP "Enter the Ellipse# (1-20)";
570  INPUT Ellipse
580  PRINT Ellipse
590  IF Ellipse<=20 AND Ellipse>0 THEN CALL Get_ell_mean(Ellipse)
600  UNTIL Ellipse=0
610  CASE "B","b" ! Get ellipse covariance matrices
620  Clearscreen
630  PRINT "Getting Shape (Covariance Matrix) for Ellipse#";
640  REPEAT
650  Ellipse=0
660  DISP "Enter the Ellipse# (1-20)";
670  INPUT Ellipse
680  PRINT Ellipse
690  IF Ellipse<=20 AND Ellipse>0 THEN CALL Get_covariance(Ellipse)
700  UNTIL Ellipse=0
710  CASE "C","c" ! Get ellipse axes and orientation

```

```

700      Clearscreen
710      PRINT "Getting Shape (Axes and Orientation) for Ellipse#";
720      REPEAT
730          Ellipse=0
740          DISP "Enter the Ellipse# (1-20)";
750          INPUT Ellipse
760          PRINT Ellipse
770          IF Ellipse<=20 AND Ellipse>0 THEN CALL Get_axes(Ellipse)
780      UNTIL Ellipse=0
790      CASE "D","d"      ! Combine ellipses
800          Clearscreen
810          PRINT "Combining two ellipses"
820          PRINT
830          Ellipse1=0
840          Ellipse2=0
850          Ellipse3=0
860          PRINT "Ellipse 1 #";
870          DISP "Enter the first Ellipse# (1-20)";
880          INPUT Ellipse1
890          PRINT Ellipse1
900          PRINT "ellipse 2 #";
910          DISP "Enter the second Ellipse# (1-20)";
920          INPUT Ellipse2
930          PRINT Ellipse2
940          PRINT "Combined Ellipse #";
950          DISP "Enter the combined Ellipse# (1-20)";
960          INPUT Ellipse3
970          PRINT Ellipse3
980          IF Ellipse1<=20 AND Ellipse1>0 AND Ellipse2<=20 AND Ellipse2>0 AND Ell
ipse3<=20 AND Ellipse3>0 AND Ellipse3<>Ellipse1 AND Ellipse3<>Ellipse2 THEN
990              Combine_ellipse(Ellipse1,Ellipse2,Ellipse3)
1000         END IF
1010         PRINT "New Mean Point: ",Xbar(Ellipse3),", ",Ybar(Ellipse3)
1020         PRINT
1030         PRINT "New Covariance Matrix:"
1040         Display_cov(Ellipse3)
1050         Pauseabit
1060     CASE "E","e"      ! Graph ellipses
1061     REPEAT
1070         Clearscreen
1080         PRINT "Graphing Ellipse#";
1090         Ellipse=0
1100         DISP "Enter Ellipse# (1-20)";
1110         INPUT Ellipse
1120         PRINT Ellipse
1130         IF Ellipse<=20 AND Ellipse>0 THEN
1140             PRINT
1150             PRINT "Center Point (",Xbar(Ellipse);", ",Ybar(Ellipse);")"
1160             PRINT
1170             PRINT "Covariance Matrix:"
1180             Display_cov(Ellipse)
1190             PRINT "Major: ",Major_axis(Ellipse);" Minor: ",Minor_axis(Ellip
se)
1200             PRINT "Orient: ",Angle(Ellipse)
1210             Invert(Ellipse,Ellipse+20)
1220             Get_bounds(Ellipse,Prob_constant)
1230             Draw_ellipse(Ellipse,Xmin,Xmax,Ymin,Ymax,Prob_constant)
1240             ALPHA ON
1250         END IF
1260     UNTIL Ellipse=0
1270     CASE "F","f"      ! Clear the graphics screen
1280     GCLEAR
1290     CASE "G","g"      ! Choose the plotter device
1300     Choose_plotter
1310     CASE "H","h"      ! Set the screen boundaries
1320     Clearscreen

```

```

1330 PRINT "Setting Screen Boundaries...."
1340 PRINT
1350 REPEAT
1360     Ellipse=0
1370     DISP "Display Extreme Points for Ellipse# (1-20)";
1380     INPUT Ellipse
1390     IF Ellipse>0 AND Ellipse<=20 THEN
1400         Invert(Ellipse,Ellipse+20)
1410         Get_bounds(Ellipse,Prob_constant)
1420         Disp_extremes(Ellipse)
1430     END IF
1440 UNTIL Ellipse=0
1450 PRINT
1460 Get_x:
1470 PRINT "Minimum X value: ";
1480 DISP "Enter the minimum X value";
1490 INPUT Xmin
1500 PRINT Xmin
1510 PRINT "Maximum X value: ";
1520 DISP "Enter the maximum X value";
1530 INPUT Xmax
1540 PRINT Xmax
1550 IF Xmin>Xmax THEN Get_x
1560 Get_y:
1570 PRINT "Minimum Y value: ";
1580 DISP "Enter the minimum Y value";
1590 INPUT Ymin
1600 PRINT Ymin
1610 PRINT "Maximum Y value: ";
1620 DISP "Enter the maximum Y value";
1630 INPUT Ymax
1640 PRINT Ymax
1650 IF Ymin>Ymax THEN Get_y
1660 CASE "I","i"      ! Calculate the chi-square acceptance test statistic
1670     Clearscreen
1680     PRINT "Calculating Acceptance Test"
1690     PRINT
1700     Ellipse1=0
1710     Ellipse2=0
1720     Swork=0
1730     PRINT "Ellipse 1 #";
1740     DISP "Enter the first Ellipse# (1-20)";
1750     INPUT Ellipse1
1760     PRINT Ellipse1
1770     PRINT "ellipse 2 #";
1780     DISP "Enter the second Ellipse# (1-20)";
1790     INPUT Ellipse2
1800     PRINT Ellipse2
1810     PRINT "Scratch Work #";
1820     DISP "Enter the scratch work # (1-20)";
1830     INPUT Swork
1840     PRINT Swork
1850     IF Ellipse1<=20 AND Ellipse1>0 AND Ellipse2<=20 AND Ellipse2>0 AND Swo
rk<=20 AND Swork>0 AND Swork<>Ellipse1 AND Swork<>Ellipse2 THEN
1860         Test(Ellipse1,Ellipse2,Swork,Test_stat)
1870     END IF
1880     PRINT
1890     PRINT "The Test statistic is ";Test_stat
1900     PRINT
1910     PRINT
1920     Pauseabit
1921 CASE "J","j"      ! Draw in the axes
1922     GRAPHICS ON
1923     CSIZE 2
1925     MOVE 0,0
1926     DRAW 0,100

```

```

1927     MOVE 100,0
1928     DRAW 0,0
1929     LABEL "(",Xmin;",",Ymin;")"
1930     LORG 7
1931     MOVE 100,0
1932     LABEL Xmax
1933     LORG 3
1934     MOVE 0,100
1935     LABEL Ymax
1936     LORG 4
1937     MOVE 50,0
1938     CSIZE 3
1940     LABEL "X-Axis (in kilometers)"
1941     MOVE 0,50
1942     DEG
1943     LDIR 270
1944     LABEL "Y-Axis (in kilometers)"
1945     LORG 1
1946     LDIR 0
1947     RAD
1949     CASE "X","x"           ! We can stop now
1950         Clearscreen
1951         PRINT "That's all, folks!"
1960         STOP
1970     CASE ELSE
1980         PRINT CHR$(7)
1990     END SELECT
2000     GOTO Menu_top
2010     END
5000     !
5010     ! SUBROUTINES: Taken from ELLIPSTUFF
5020     !
5030     SUB Invert(Srce, Dest)
5040         ! This routine inverts any covariance matrix in Matrx and places
5050         ! the inverted matrix in Dest.
5060         COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
5070         Det=Matrx(1,1,Srce)*Matrx(2,2,Srce)-Matrx(1,2,Srce)*Matrx(1,2,Srce)
5080         Matrx(1,1, Dest)=Matrx(2,2,Srce)/Det
5090         Matrx(2,2, Dest)=Matrx(1,1,Srce)/Det
5100         Matrx(1,2, Dest)=-Matrx(1,2,Srce)/Det
5110         Matrx(2,1, Dest)=-Matrx(2,1,Srce)/Det
5120     SUBEND ! End of SUB Invert
5130     !
5140     ! GET_BOUNDS
5150     !
5160     SUB Get_bounds(Ellipse, Prob_constant)
5170         ! This subroutine calculates the X and Y limits for the given ellipse
.
5180         COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
5190         REAL Temp1,Temp2,Temp3,Temp4,Temp5
5210         Temp1=Matrx(2,2,Ellipse+20)*Prob_constant
5220         Temp2=Matrx(1,2,Ellipse+20)*Matrx(1,2,Ellipse+20)-Matrx(1,1,Ellipse+20
)*Matrx(2,2,Ellipse+20)
5230         Temp3=(Temp1/Temp2)^(.5)
5240         Temp4=Matrx(1,1,Ellipse+20)*Prob_constant
5250         Temp5=(Temp4/Temp2)^(.5)
5260         Emin_y(Ellipse)=-Temp5*Ybar(Ellipse)
5270         Emax_y(Ellipse)=Temp5*Ybar(Ellipse)
5280         Emin_x(Ellipse)=-Temp3*Xbar(Ellipse)
5290         Emax_x(Ellipse)=Temp3*Xbar(Ellipse)
5300     SUBEND ! End of GET_BOUNDS
5310     !
5320     ! CHOOSE_PLOTTER
5330     !

```

```

5340 SUB Choose_plotter
5350 ! Subroutine to choose the desired plotter device
5360 REPEAT
5370   Go on=-1
5380   DISP "Do you want to (D)raw, (E)rase, Plot with Pen (0), (1), (2)?"
5390   !
5400   LINPUT Message$
5410   SELECT Message$
5420   CASE "D","d"
5430     PLOTTER IS 3,"INTERNAL"
5440     GRAPHICS ON
5450     PEN 1
5460     Go on=1
5470   CASE "0"
5480     PLOTTER IS 705,"HPGL"
5490     GRAPHICS ON
5500     PEN 0
5510     Go on=1
5520   CASE "1"
5530     PLOTTER IS 705,"HPGL"
5540     GRAPHICS ON
5550     PEN 1
5560     Go on=1
5570   CASE "2"
5580     PLOTTER IS 705,"HPGL"
5590     GRAPHICS ON
5600     PEN 2
5610     Go on=1
5620   CASE "E","e"
5630     PLOTTER IS 3,"INTERNAL"
5640     GRAPHICS ON
5650     PEN -1
5660     Go on=1
5670   CASE ELSE
5680     PRINT CHR$(7)
5690   END SELECT
5700 UNTIL Go_on=1
5710 SUBEND
5720 !
5730 ! GET_ELL_MEAN
5740 !
5750 SUB Get_ell_mean(Ellipse)
5760 ! This subroutine prompts the user for the mean of an ellipse.
5770 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
5780 is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
5790 Clearscreen
5800 PRINT TABXY(1,10);"WHAT IS X-BAR for ELLIPSE ";Ellipse;"?";
5810 INPUT Xbar(Ellipse)
5820 PRINT TABXY(1,11);"WHAT IS Y-BAR for ELLIPSE ";Ellipse;"?";
5830 INPUT Ybar(Ellipse)
5840 PRINT Ybar(Ellipse)
5850 Pauseabit
5860 SUBEND
5870 !
5880 ! GET_COVARIANCE
5890 !
5900 SUB Get_covariance(Ellipse)
5910 ! This routine gets the covariance matrix for an ellipse
5920 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
5930 is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
5940 Clearscreen
5950 FOR K=1 TO 2
5960   PRINT TABXY(1,3+K*6);"ENTER ELEMENT (";K;" ";K;" ) IN THE COVARIANCE
5970   MATRIX FOR ELLIPSE#";Ellipse;" ";
5980   INPUT Matrx(K,K,Ellipse)

```

```

5860         PRINT Matrx(K,K,Ellipse)
5870     NEXT K
5880     PRINT TABXY(1,3+9);"ENTER ELEMENT (1,2) IN THE COVARIANCE MATRIX FOR E
LEMENT#";I;": ";
5890     INPUT Matrx(1,2,Ellipse)
5900     Matrx(2,1,Ellipse)=Matrx(1,2,Ellipse)
5910     PRINT Matrx(1,2,Ellipse)
5920     Pauseabit
5930 SUBEND
5940 !
5950 ! PAUSEABIT
5960 !
5970 SUB Pauseabit
5980     ! Pause and wait for a carriage return
5990     DISP "Type ENTER to continue...";
6000     INPUT Garbage$
6010 SUBEND
6020 SUB Clearscreen
6030     ! Clear the screen
6040     PRINT CHR$(12)
6050 SUBEND
6060 !
6070 ! GET_AXES
6080 !
6090 SUB Get_axes(Ellipse)
6100     ! This subroutine gets an ellipse in terms of the axes and the
6110     ! angle of orientation. These are converted into a covariance
6120     ! matrix.
6130     COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
6140     Prob_constant=2*LOG(.05)
6150     Clearscreen
6160     PRINT "WHAT IS THE RADIUS OF THE MAJOR AXIS OF ELLIPSE ";Ellipse;" : "
;
6170     INPUT Major_axis(Ellipse)
6180     PRINT Major_axis(Ellipse)
6190     PRINT "WHAT IS THE RADIUS OF THE MINOR AXIS OF ELLIPSE ";Ellipse;" : "
;
6200     INPUT Minor_axis(Ellipse)
6210     PRINT Minor_axis(Ellipse)
6220     PRINT "WHAT IS THE ANGLE (IN DEGREES) OF THE MAJOR AXIS OF ELLIPSE ";E
llipse;" : ";
6230     INPUT Angle(Ellipse)
6240     PRINT Angle(Ellipse)
6250     ! Convert to Covariance Matrix
6260     Theta(Ellipse)=PI*Angle(Ellipse)/180
6270     Matrx(1,1,Ellipse)=(Major_axis(Ellipse)*COS(Theta(Ellipse)))^2+(Minor_
axis(Ellipse)*SIN(Theta(Ellipse)))^2
6280     Matrx(2,2,Ellipse)=(Major_axis(Ellipse)*SIN(Theta(Ellipse)))^2+(Minor_
axis(Ellipse)*COS(Theta(Ellipse)))^2
6290     Matrx(2,1,Ellipse)=((Major_axis(Ellipse))^2-(Minor_axis(Ellipse))^2)*C
OS(Theta(Ellipse))*SIN(Theta(Ellipse))
6300     Matrx(1,2,Ellipse)=Matrx(2,1,Ellipse)
6310     FOR J=1 TO 2
6320         FOR K=1 TO 2
6330             Matrx(J,K,Ellipse)=Matrx(J,K,Ellipse)/(-Prob_constant)
6340         NEXT K
6350     NEXT J
6360     PRINT
6370     Pauseabit
6380 SUBEND
6390 !
6400 ! DISPLAY_COV
6410 !
6420 SUB Display_cov(Ellipse)
6430     ! Displays the covariance matrix for an ellipse

```

```

6440      COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
6450      FOR I=1 TO 2
6460          FOR J=1 TO 2
6470              PRINT Matrx(I,J,Ellipse);" ";
6480          NEXT J
6490      PRINT
6500  NEXT I
6510  SUBEND
6520  !
6530  ! ADD_COVARIANCE
6540  !
6550  SUB Add_covariance(First,Second,Summer)
6560      ! This subroutine can be used to add Summer=First+Second
6570      COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
6580      INTEGER I,J
6590      FOR I=1 TO 2
6600          FOR J=1 TO 2
6610              Matrx(I,J,Summer)=Matrx(I,J,First)+Matrx(I,J,Second)
6620          NEXT J
6630      NEXT I
6640  SUBEND
6650  !
6660  ! COMBINE_ELLIPSE
6670  !
6680  SUB Combine_ellipse(First,Second,Combo)
6690      ! This routine finds the "JPL" combination of the First and Second
! ellipses.
6710      COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
6720      REAL Sx1,Sx2
6730      ! Calculate the new covariance matrix.
6740      Invert(First,First+20)
6750      Invert(Second,Second+20)
6760      Add_covariance(First+20,Second+20,Combo+20)
6770      Invert(Combo+20,Combo)
6780      ! Calculate the new mean point.
6790      Sx1=Matrx(1,1,First+20)*Xbar(First)+Matrx(1,2,First+20)*Ybar(First)
6800      Sx1=Sx1+Matrx(1,1,Second+20)*Xbar(Second)+Matrx(1,2,Second+20)*Ybar(Se
cond)
6810      Sx2=Matrx(2,1,First+20)*Xbar(First)+Matrx(2,2,First+20)*Ybar(First)
6820      Sx2=Sx2+Matrx(2,1,Second+20)*Xbar(Second)+Matrx(2,2,Second+20)*Ybar(Se
cond)
6830      Xbar(Combo)=Matrx(1,1,Combo)*Sx1+Matrx(1,2,Combo)*Sx2
6840      Ybar(Combo)=Matrx(2,1,Combo)*Sx1+Matrx(2,2,Combo)*Sx2
6850  SUBEND
6860  !
6870  ! DISP_EXTREMES
6880  !
6890  SUB Disp_extremes(Ellipse)
6900      ! This routine displays the extreme x and y values for the specified
! ellipse
6920      COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
6930      PRINT "E# ";Ellipse;" ";
6940      PRINT "Xmin ";Emin_x(Ellipse);
6950      PRINT " "; Xmax ";Emax_x(Ellipse);
6960      PRINT " "; Ymin ";Emin_y(Ellipse);
6970      PRINT " "; Ymax ";Emax_y(Ellipse)
6980  SUBEND
6990  SUB Draw_ellipse(Ellipse,Xmin,Xmax,Ymin,Ymax,Prob_constant)
7000      ! This routine draws one (1) ellipse on the current plotter device.
7010      COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
7020      INTEGER Sign

```

```

7030     REAL Xpoint,Upoint,Ypoint,Temp1,Temp2,Temp3,Temp4,Temp5
7060     GRAPHICS ON
7070     ! Draw top half of the ellipse
7080     Sign=1
7090     GOSUB Draw_half
7100     ! Draw bottom half of the ellipse
7110     Sign=-1
7120     GOSUB Draw_half
7130     ! Finish up
7140     MOVE (Xbar(Ellipse)-Xmin)*100/(Xmax-Xmin),(Ybar(Ellipse)-Ymin)*100/(Ym
ax-Ymin)
7150     DRAW (Xbar(Ellipse)-Xmin)*100/(Xmax-Xmin),(Ybar(Ellipse)-Ymin)*100/(Ym
ax-Ymin)
7160     ! Okey, dokey, we're done here.
7170     SUBEXIT
7180 Draw_half: ! Draw half of the ellipse
7190     ! If Sign=1, then draw top half; if Sign=-1, then bottom half.
7200     FOR Xpoint=Emin_x(Ellipse) TO Emax_x(Ellipse) STEP .1
7210         GOSUB Draw_1_point
7220     NEXT Xpoint
7230     Xpoint=Emax_x(Ellipse)
7240     GOSUB Draw_1_point
7250     RETURN
7260     !
7270     ! Compute each point and draw the new line. (It's here since we call
7280     ! it twice
7290     !
7300 Draw_1_point: !
7310     Upoint=Xpoint-Xbar(Ellipse)
7320     Temp1=Matrx(1,2,Ellipse+20)*Upoint
7330     Temp2=Temp1*Temp1-Matrx(2,2,Ellipse+20)*(Matrx(1,1,Ellipse+20)*Upoint*
Upoint+Prob_constant)
7340     IF Temp2<10^(-10) THEN Temp2=0
7350     Ypoint=(-Temp1+Sign*SQR(Temp2))/Matrx(2,2,Ellipse+20)+Ybar(Ellipse)
7360     IF Xpoint<Emin_x(Ellipse)+.001 THEN
7370         MOVE (Xpoint-Xmin)*100/(Xmax-Xmin),(Ypoint-Ymin)*100/(Ymax-Ymin)
7380     ELSE
7390         DRAW (Xpoint-Xmin)*100/(Xmax-Xmin),(Ypoint-Ymin)*100/(Ymax-Ymin)
7400     END IF
7410     RETURN
7420 SUBEND
7430 !
7440 ! TEST
7450 !
7460 SUB Test(First,Second,Work,Test_val)
7470     ! This routines calculates the acceptance test criteria for First and
7480     ! Second. Work is used as a "scratchpad".
7490     COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
7500     Add_covariance(First,Second,Work)
7510     Invert(Work,Work+20)
7520     Diffx=Xbar(First)-Xbar(Second)
7530     Diffy=Ybar(First)-Ybar(Second)
7540     Temp1=Matrx(1,1,Work+20)*Diffx+Matrx(1,2,Work+20)*Diffy
7550     Temp2=Matrx(2,1,Work+20)*Diffx+Matrx(2,2,Work+20)*Diffy
7560     Test_val=Diffx*Temp1+Diffy*Temp2
7570 SUBEND

```

B. ELLSIM: Ellipse Simulation Program

Ellsim is a "confidence ellipse simulation" program. It has been used to explore the robustness and properties of the statistical test (which is used to decide whether or not to combine two ellipses). In it, two normal data distributions are specified, corresponding to two emitters. Confidence ellipses are generated, tested, and combined, and various descriptive statistics are compiled. In addition, there is a routine to calculate the power of the statistical test in certain cases.

```

1000 |-----|
1010 |           Ellipse Simulation Driver           |
1020 |-----|
1030 |           This is a simple simulation program for the ellipse |
1040 |           combination test. Only the Normal distribution is  |
1050 |           supported.                                         |
1060 |-----|
1070 |           Original:  1/17/85                      Updated:  7/29/85 |
1080 |-----|
1090 |
1100 |
1110 | 7/29/85
1120 Top_of_program:  !
1130 | Use Fast Math card
1140 CONTROL 32,2;1
1150 | Specify common variables between the different subprograms
1160 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axi
s(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
1170 COM /Driver/ Obs(2)
1180 Driver_menu:  !
1190 Clearscreen
1200 PRINT "Ellipse Combination Program Driver"
1210 PRINT
1220 PRINT "(A) -- Specify True Covariance Matrices"
1230 PRINT "(B) -- Specify True Mean Parameters"
1240 PRINT "(C) -- Specify Observations"
1250 PRINT "(D) -- Call Simulation Generator"
1260 PRINT "(E) -- Call Power Generator"
1270 PRINT "(X) -- Exit Program"
1280 PRINT
1290 Get_choice:  !
1300 INPUT "Enter your choice:",Option$
1310 SELECT Option$
1320 CASE "A","a"
1330 Clearscreen
1340 CALL Get_covariance(1)
1350 CALL Get_covariance(2)
1360 CASE "B","b"
1370 Clearscreen
1380 CALL Get_ell_mean(1)
1390 CALL Get_ell_mean(2)
1400 CASE "C","c"
1410 Clearscreen
1420 GOSUB Get_obs
1430 CASE "D","d"
1440 Clearscreen
1450 CALL Sim_ellipse
1460 CASE "E","e"
1470 Clearscreen
1480 CALL Power_ellipse
1490 CASE "X","x"
1500 GOTO End_program
1510 CASE ELSE
1520 PRINT CHR$(7)
1530 GOTO Get_choice
1540 END SELECT
1550 GOTO Driver_menu
1560 |-----|
1570 |           SUBROUTINES           |
1580 |-----|
1590 |
1600 | Get the number of TRIALS (Main Menu Choice "B")
1610 |
1620 Get_obs:  !
1630 FOR Imat=1 TO 2

```

```

1640     PRINT "How many observations for ellipse #";Imat;": ";
1650     INPUT Obs(Imat)
1660     PRINT Obs(Imat)
1670     NEXT Imat
1680 RETURN
1690 End_program:
1700     END
1710     !-----!
1720     ! SUBPROGRAMS AND FUNCTIONS
1730     !
1740     SUB Invert(Srce, Dest)
1750     ! This routine inverts any covariance matrix in Matrx and places
1760     ! the inverted matrix in Dest.
1770     COM /Ellipses/ Matrx(2,2,40), Xbar(20), Ybar(20), Major_axis(20), Minor_ax
1780     is(20), Theta(20), Angle(20), Emin_x(20), Emax_x(20), Emin_y(20), Emax_y(20)
1790     Det=Matrx(1,1,Srce)*Matrx(2,2,Srce)-Matrx(1,2,Srce)*Matrx(1,2,Srce)
1800     Matrx(1,1, Dest)=Matrx(2,2,Srce)/Det
1810     Matrx(2,2, Dest)=Matrx(1,1,Srce)/Det
1820     Matrx(1,2, Dest)=-Matrx(1,2,Srce)/Det
1830     Matrx(2,1, Dest)=-Matrx(2,1,Srce)/Det
1840     SUBEND ! End of SUB Invert
1850     !
1860     ! GET_BOUNDS
1870     SUB Get_bounds(Ellipse, Prob_constant)
1880     ! This subroutine calculates the X and Y limits for the given ellipse
1890     COM /Ellipses/ Matrx(2,2,40), Xbar(20), Ybar(20), Major_axis(20), Minor_ax
1900     is(20), Theta(20), Angle(20), Emin_x(20), Emax_x(20), Emin_y(20), Emax_y(20)
1910     REAL Temp1, Temp2, Temp3, Temp4, Temp5
1920     Temp1=Matrx(2,2, Ellipse+20)*Prob_constant
1930     Temp2=Matrx(1,2, Ellipse+20)*Matrx(1,2, Ellipse+20)-Matrx(1,1, Ellipse+20
1940     )*Matrx(2,2, Ellipse+20)
1950     Temp3=(Temp1/Temp2)^(.5)
1960     Temp4=Matrx(1,1, Ellipse+20)*Prob_constant
1970     Temp5=(Temp4/Temp2)^(.5)
1980     Emin_y(Ellipse)=-Temp5+Ybar(Ellipse)
1990     Emax_y(Ellipse)=Temp5+Ybar(Ellipse)
2000     Emin_x(Ellipse)=-Temp3+Xbar(Ellipse)
2010     Emax_x(Ellipse)=Temp3+Xbar(Ellipse)
2020     SUBEND ! End of GET_BOUNDS
2030     !
2040     ! CHOOSE_PLOTTER
2050     SUB Choose_plotter
2060     ! Subroutine to choose the desired plotter device
2070     REPEAT
2080     Go_on=-1
2090     DISP "Do you want to (D)raw, (E)rase, Plot with Pen (0), (1), (2)?"
2100     !
2110     INPUT Message$
2120     SELECT Message$
2130     CASE "D", "d"
2140     PLOTTER IS 3, "INTERNAL"
2150     GRAPHICS ON
2160     PEN 1
2170     Go_on=1
2180     CASE "E", "e"
2190     PLOTTER IS 705, "HPGL"
2200     GRAPHICS ON
2210     PEN 0
2220     Go_on=1
2230     CASE "P", "p"
2240     PLOTTER IS 705, "HPGL"
2250     GRAPHICS ON
2260     PEN 1

```

```

2250         Go_on=1
2260     CASE "2"
2270         PLOTTER IS 705,"HPGL"
2280         GRAPHICS ON
2290         PEN 2
2300         Go_on=1
2310     CASE "E","e"
2320         PLOTTER IS 3,"INTERNAL"
2330         GRAPHICS ON
2340         PEN -1
2350         Go_on=1
2360     CASE ELSE
2370         PRINT CHR$(7)
2380     END SELECT
2390 UNTIL Go_on=1
2400 SUBEND
2410 !
2420 ! GET_ELL_MEAN
2430 !
2440 SUB Get_ell_mean(Ellipse)
2450     ! This subroutine prompts the user for the mean of an ellipse.
2460     COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
2470     Clearscreen
2480     PRINT TABXY(1,10);"WHAT IS X-BAR for ELLIPSE ";Ellipse;"?";
2490     INPUT Xbar(Ellipse)
2500     PRINT Xbar(Ellipse)
2510     PRINT TABXY(1,11);"WHAT IS Y-BAR for ELLIPSE ";Ellipse;"?";
2520     INPUT Ybar(Ellipse)
2530     PRINT Ybar(Ellipse)
2540     Pauseabit
2550 SUBEND
2560 !
2570 ! GET_COVARIANCE
2580 !
2590 SUB Get_covariance(Ellipse)
2600     ! This routine gets the covariance matrix for an ellipse
2610     COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
2620     Clearscreen
2630     FOR K=1 TO 2
2640         PRINT TABXY(1,3+K*6);"ENTER ELEMENT (";K;" ";K;" ) IN THE COVARIANCE
MATRIX FOR ELLIPSE";Ellipse;" ";
2650         INPUT Matrx(K,K,Ellipse)
2660         PRINT Matrx(K,K,Ellipse)
2670     NEXT K
2680     PRINT TABXY(1,3+9);"ENTER ELEMENT (1,2) IN THE COVARIANCE MATRIX FOR E
LEMENT";1;" ";
2690     INPUT Matrx(1,2,Ellipse)
2700     Matrx(2,1,Ellipse)=Matrx(1,2,Ellipse)
2710     PRINT Matrx(1,2,Ellipse)
2720     Pauseabit
2730 SUBEND
2740 !
2750 ! PAUSEABIT
2760 !
2770 SUB Pauseabit
2780     ! Pause and wait for a carriage return
2790     DISP "Type ENTER to continue...";
2800     INPUT Garbage$
2810 SUBEND
2820 SUB Clearscreen
2830     ! Clear the screen
2840     PRINT CHR$(12)
2850 SUBEND
2860 !

```

```

2870 ! GET_AXES
2880 !
2890 SUB Get_axes(Ellipse)
2900 ! This subroutine gets an ellipse in terms of the axes and the
2910 ! angle of orientation. These are converted into a covariance
2920 ! matrix.
2930 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
2940 Prob_constant=2*LOG(.05)
2950 Clearscreen
2960 PRINT "WHAT IS THE RADIUS OF THE MAJOR AXIS OF ELLIPSE ";Ellipse;" : "
;
2970 INPUT Major_axis(Ellipse)
2980 PRINT Major_axis(Ellipse)
2990 PRINT "WHAT IS THE RADIUS OF THE MINOR AXIS OF ELLIPSE ";Ellipse;" : "
;
3000 INPUT Minor_axis(Ellipse)
3010 PRINT Minor_axis(Ellipse)
3020 PRINT "WHAT IS THE ANGLE (IN DEGREES) OF THE MAJOR AXIS OF ELLIPSE ";E
llipse;" : ";
3030 INPUT Angle(Ellipse)
3040 PRINT Angle(Ellipse)
3050 ! Convert to Covariance Matrix
3060 Theta(Ellipse)=PI*Angle(Ellipse)/180
3070 Matrx(1,1,Ellipse)=(Major_axis(Ellipse)*COS(Theta(Ellipse)))^2+(Minor_
axis(Ellipse)*SIN(Theta(Ellipse)))^2
3080 Matrx(2,2,Ellipse)=(Major_axis(Ellipse)*SIN(Theta(Ellipse)))^2+(Minor_
axis(Ellipse)*COS(Theta(Ellipse)))^2
3090 Matrx(2,1,Ellipse)=((Major_axis(Ellipse))^2-(Minor_axis(Ellipse))^2)*C
OS(Theta(Ellipse))*SIN(Theta(Ellipse))
3100 Matrx(1,2,Ellipse)=Matrx(2,1,Ellipse)
3110 FOR J=1 TO 2
3120   FOR K=1 TO 2
3130     Matrx(J,K,Ellipse)=Matrx(J,K,Ellipse)/(-Prob_constant)
3140   NEXT K
3150 NEXT J
3160 PRINT
3170 Pauseabit
3180 SUBEND
3190 !
3200 ! DISPLAY_COV
3210 !
3220 SUB Display_cov(Ellipse)
3230 ! Displays the covariance matrix for an ellipse
3240 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
3250 FOR I=1 TO 2
3260   FOR J=1 TO 2
3270     PRINT Matrx(I,J,Ellipse);"  ";
3280   NEXT J
3290   PRINT
3300 NEXT I
3310 SUBEND
3320 !
3330 ! ADD_COVARIANCE
3340 !
3350 SUB Add_covariance(First,Second,Summer)
3360 ! This subroutine can be used to add Summer=First+Second
3370 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
3380 INTEGER I,J
3390 FOR I=1 TO 2
3400   FOR J=1 TO 2
3410     Matrx(I,J,Summer)=Matrx(I,J,First)+Matrx(I,J,Second)
3420   NEXT J
3430 NEXT I

```

```

3440 SUBEND
3450 !
3460 ! COMBINE_ELLIPSE
3470 !
3480 SUB Combine_ellipse(First,Second,Combo)
3490 ! This routine finds the "JPL" combination of the First and Second
3500 ! ellipses.
3510 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
3520 is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
3520 REAL Sx1,Sx2
3530 ! Calculate the new covariance matrix.
3540 Invert(First,First+20)
3550 Invert(Second,Second+20)
3560 Add_covariance(First+20,Second+20,Combo+20)
3570 Invert(Combo+20,Combo)
3580 ! Calculate the new mean point.
3590 Sx1=Matrx(1,1,First+20)*Xbar(First)+Matrx(1,2,First+20)*Ybar(First)
3600 Sx1=Sx1+Matrx(1,1,Second+20)*Xbar(Second)+Matrx(1,2,Second+20)*Ybar(Se
cond)
3610 Sx2=Matrx(2,1,First+20)*Xbar(First)+Matrx(2,2,First+20)*Ybar(First)
3620 Sx2=Sx2+Matrx(2,1,Second+20)*Xbar(Second)+Matrx(2,2,Second+20)*Ybar(Se
cond)
3630 Xbar(Combo)=Matrx(1,1,Combo)*Sx1+Matrx(1,2,Combo)*Sx2
3640 Ybar(Combo)=Matrx(2,1,Combo)*Sx1+Matrx(2,2,Combo)*Sx2
3650 SUBEND
3660 !
3670 ! DISP_EXTREMES
3680 !
3690 SUB Disp_extremes(Ellipse)
3700 ! This routine displays the extreme x and y values for the specified
3710 ! ellipse
3720 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
3730 is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
3730 PRINT "E# ";Ellipse;": ";
3740 PRINT "Xmin ";Emin_x(Ellipse);
3750 PRINT "Xmax ";Emax_x(Ellipse);
3760 PRINT "Ymin ";Emin_y(Ellipse);
3770 PRINT "Ymax ";Emax_y(Ellipse)
3780 SUBEND
3790 SUB Draw_ellipse(Ellipse,Xmin,Xmax,Ymin,Ymax,Prob_constant)
3800 ! This routine draws one (1) ellipse on the current plotter device.
3810 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
3820 is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
3820 INTEGER Sign
3830 REAL Xpoint,Upoint,Ypoint,Temp1,Temp2,Temp3,Temp4,Temp5
3840 GRAPHICS ON
3850 ! Draw top half of the ellipse
3860 Sign=1
3870 GOSUB Draw_half
3880 ! Draw bottom half of the ellipse
3890 Sign=-1
3900 GOSUB Draw_half
3910 ! Finish up
3920 MOVE (Xbar(Ellipse)-Xmin)*100/(Xmax-Xmin),(Ybar(Ellipse)-Ymin)*100/(Ym
ax-Ymin)
3930 DRAW (Xbar(Ellipse)-Xmin)*100/(Xmax-Xmin),(Ybar(Ellipse)-Ymin)*100/(Ym
ax-Ymin)
3940 ! Okay, dokey, we're done here.
3950 SUBEXIT
3960 Draw_half: ! Draw half of the ellipse
3970 ! If Sign=1, then draw top half; if Sign=-1, then bottom half.
3980 FOR Xpoint=Emin_x(Ellipse) TO Emax_x(Ellipse) STEP .1
3990 GOSUB Draw_1_point
4000 NEXT Xpoint
4010 Xpoint=Emax_x(Ellipse)
4020 GOSUB Draw_T_point

```

```

4030      RETURN      --
4040      !
4050      !   Compute each point and draw the new line.  (It's here since we call
4060      !   it twice
4070      !
4080      Draw 1 point: 1
4090      Upoint=Xpoint-Xbar(Ellipse)
4100      Temp1=Matrx(1,2,Ellipse+20)*Upoint
4110      Temp2=Temp1*Temp1-Matrx(2,2,Ellipse+20)*(Matrx(1,1,Ellipse+20)*Upoint+
Upoint*Prob constant)
4120      IF Temp2<10^(-10) THEN Temp2=0
4130      Ypoint=(-Temp1+Sign*SQR(Temp2))/Matrx(2,2,Ellipse+20)+Ybar(Ellipse)
4140      IF Xpoint<=Emin_x(Ellipse)+.001 THEN
4150          MOVE (Xpoint-Xmin)*100/(Xmax-Xmin),(Ypoint-Ymin)*100/(Ymax-Ymin)
4160      ELSE
4170          DRAW (Xpoint-Xmin)*100/(Xmax-Xmin),(Ypoint-Ymin)*100/(Ymax-Ymin)
4180      END IF
4190      RETURN
4200      SUBEND
4210      !
4220      ! TEST
4230      !
4240      SUB Test(First,Second,Work,Test_val)
4250          ! This routine calculates the acceptance test criteria for First and
4260          ! Second.  Work is used as a "scratchpad".
4270          COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
4280          Add_covariance(First,Second,Work)
4290          Invert(Work,Work+20)
4300          Diffx=Xbar(First)-Xbar(Second)
4310          Diffy=Ybar(First)-Ybar(Second)
4320          Temp1=Matrx(1,1,Work+20)*Diffx+Matrx(1,2,Work+20)*Diffy
4330          Temp2=Matrx(2,1,Work+20)*Diffx+Matrx(2,2,Work+20)*Diffy
4340          Test_val=Diffx*Temp1+Diffy*Temp2
4350      SUBEND
4360      SUB Sim ellipse
4370          ! Ellipse Combination Simulation Program
4380          ! 1/16/85   Update 7/18/85
4390          !
4400          !   Specify the common variables
4410          !
4420          COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
4430          COM /Driver/ Obs(2)
4440          ! Variable Definitions
4450          !
4460          !
4470          INTEGER Ellnum,Imat,Xmin,Xmax,Ymin,Ymax
4480          INTEGER Mobervations(2)
4490          DIM Sx(2,10),Xs(5000),Ys(5000)
4500          !
4510          !   Use of MATRIX Array
4520          !
4530          ! MATRIX 1 is an initial matrix
4540          ! MATRIX 2 is an initial matrix
4550          ! MATRIX 3 is the theoretical combination of 8 and 9
4560          ! MATRIX 4 is the estimate of 8
4570          ! MATRIX 5 is the estimate of 9
4580          ! MATRIX 6 is the combination of 4 and 5
4590          ! MATRIX 7 is the sum of 3 and 4 (for test)
4600          ! Note that if the True Variance-Covariance is used instead of the
4610          ! the estimate, this is equal to the sum of 8 and 9.
4620          ! MATRIX 8 is MATRIX 1 divided by sample size
4630          ! MATRIX 9 is MATRIX 2 divided by sample size
4640          ! MATRIX 21-29 are the inverses of 1-9
4650          !
4660          !
4670          !
4680          !   Initialization

```

```

4690      !
4700      GOSUB Init_sub1
4710      !
4720      ! Main Program Loop
4730      !
4740 Main_loop: !
4750      Clearscreen
4760      PRINT "Ellipse Simulations"
4770      PRINT
4780      PRINT "Enter the Letter of your choice:"
4790      PRINT "      (A) Reset the Random Number Seed"
4800      PRINT "      (B) Simulate using true Covariance Matrices"
4810      PRINT "      (C) Simulate using estimated Covariance Matrices"
4820      PRINT "      (D) Display Results on the Printer"
4830      PRINT "      (X) Exit Program"
4831 Get_option: !
4840      INPUT "Enter Your Choice:",Mainchoice$
4860      SELECT Mainchoice$
4870          CASE "A","a"
4880              GOSUB Get_seed
4890          CASE "B","b"
4900              Use_true=1
4910              GOSUB Simulate
4920          CASE "C","c"
4930              Use_true=0
4940              GOSUB Simulate
4950          CASE "D","d"
4960              GOSUB Display_results
4970          CASE "X","x"
4980              GOTO End_sub1
4990          CASE ELSE
5000              PRINT CHR$(7)
5010              GOTO Get_option
5020      END SELECT
5030      GOTO Main_loop
5040      !-----!
5050      ! Utility Subroutines !
5060      !-----!
5070      !
5080      ! Initialize Program
5090      !
5100 Init_sub1: !
5101      RANDOMIZE
5110      Prob constant=2*LOG(.05)      ! Confidence level parameter
5111      FOR I=1 TO 2
5112          FOR J=1 TO 2
5113              Matrx(I,J,8)=Matrx(I,J,1)/Obs(1)
5114              Matrx(I,J,9)=Matrx(I,J,2)/Obs(2)
5115          NEXT J
5116      NEXT I
5117      Xbar(8)=Xbar(1)
5118      Ybar(8)=Ybar(1)
5119      Xbar(9)=Xbar(2)
5120      Ybar(9)=Ybar(2)
5122      ! Set parameters
5230 RETURN
5340      !
5350      ! Generate the new ellipses using random observations
5360      !
5370 Generate new: !***FLAG***
5380      FOR Imat=1 TO 2
5390          M=Obs(Imat)
5400          ! Generate the X and Y values
5410          Xsum=0
5420          Ysum=0
5430          Cc=SQR(Matrx(1,1,Imat))

```

```

5440      A=Sigma(1,2,Imat)/Cc
5450      Bee=SQR(Matrxx(2,2,Imat)-A*A)
5460      FOR I=1 TO M
5470          U1=RND
5480          U2=RND
5490          X1=SQR(-2*LOG(U1))*SIN(2*PI*U2)
5500          Y1=SQR(-2*LOG(U2))*COS(2*PI*U1)
5510          Xs(I)=Cc*X1+Xbar(Imat)
5520          Ys(I)=A*X1+Bee*Y1+Ybar(Imat)
5530          Xsum=Xsum+Xs(I)
5540          Ysum=Ysum+Ys(I)
5550      NEXT I
5560      Xbar(Imat+3)=Xsum/M
5570      Ybar(Imat+3)=Ysum/M
5580      ! Calculate the variances
5590      Matrx(1,1,Imat+3)=0
5600      Matrx(2,2,Imat+3)=0
5610      Matrx(1,2,Imat+3)=0
5620      FOR I=1 TO M
5630          Matrx(1,1,Imat+3)=Matrx(1,1,Imat+3)+(Xs(I)-Xbar(Imat+3))*(Xs(I)-Xbar(Imat+3))
5640          Matrx(2,2,Imat+3)=Matrx(2,2,Imat+3)+(Ys(I)-Ybar(Imat+3))*(Ys(I)-Ybar(Imat+3))
5650          Matrx(1,2,Imat+3)=Matrx(1,2,Imat+3)+(Xs(I)-Xbar(Imat+3))*(Ys(I)-Ybar(Imat+3))
5660      NEXT I
5670      Matrx(1,1,Imat+3)=Matrx(1,1,Imat+3)/((M-1)*M)
5680      Matrx(2,2,Imat+3)=Matrx(2,2,Imat+3)/((M-1)*M)
5690      Matrx(1,2,Imat+3)=Matrx(1,2,Imat+3)/((M-1)*M)
5700      Matrx(2,1,Imat+3)=Matrx(1,2,Imat+3)
5710  NEXT Imat
5720 RETURN ! Generate_new
5730 !
5740 !   Generate the inverses of our matrices
5750 !
5760 Gen_inverses: !
5830 Patch1: !
5840 IF Use_true=0 THEN Patch2
5850 ! To use the true matrices, copy them from 8 and 9 into 4 and 5
5860 FOR Imat=8 TO 9
5870     FOR I=1 TO 2
5880         FOR J=1 TO 2
5890             Matrx(I,J,Imat-4)=Matrx(I,J,Imat)
5900         NEXT J
5910     NEXT I
5920 NEXT Imat
5930 Patch2: !
6151 Combine_ellipse(8,9,3)
6152 Combine_ellipse(4,5,6)
6170 RETURN ! Gen_inverses
6430 !-----!
6440 !           Program Subroutines           !
6450 !-----!
6460 !
6470 ! GET SEED (Menu Menu Choice "C")
6480 !
6490 Get_seed: !
6500 Clearscreen
6510 PRINT "ENTER A SEED (1 to 2^31-2): ";
6520 INPUT Seed
6530 Seed$=VAL$(Seed)
6540 PRINT Seed$
6550 RANDOMIZE Seed
6560 PRINT
6570 Pauseabit
6580 RETURN

```

```

6590      !
6600      ! SIMULATE
6610      !
6620      Simulate:      !
6630      Clearscreen
6640      PRINT TABXY(1,17);"***** HOW MANY SIMULATIONS DO YOU WISH TO R
UN? ";
6650      INPUT Nsimulations
6660      PRINT Nsimulations
6670      N_accept=0
6680      N_reject=0
6690      Anotin1=0
6700      Anotin2=0
6710      Anotinb=0
6720      Rnotin1=0
6730      Rnotin2=0
6740      Rnotinb=0
6750      Pauseabit
6760      Clearscreen
6770      PRINT TABXY(5,3);"NUMBER OF SIMULATIONS= 0"
6780      PRINT TABXY(5,4);"*****"
6790      PRINT TABXY(5,5);"ACCEPTED ERROR ELLIPSES= 0"
6800      PRINT TABXY(5,6);"REJECTED ERROR ELLIPSES= 0"
6810      PRINT TABXY(5,9);"ACCEPTED BREAKOUT"
6820      PRINT TABXY(5,10);"Target One is NOT in Combined Ellipse= 0"
6830      PRINT TABXY(5,11);"Target Two is NOT in Combined Ellipse= 0"
6840      PRINT TABXY(5,12);"Neither Target is in Combined Ellipse= 0"
6850      PRINT TABXY(5,14);"REJECTED BREAKOUT"
6860      PRINT TABXY(5,15);"Target One is NOT in Combined Ellipse= 0"
6870      PRINT TABXY(5,16);"Target Two is NOT in Combined Ellipse= 0"
6880      PRINT TABXY(5,17);"Neither Target is in Combined Ellipse= 0"
6890      FOR Isim=1 TO Nsimulations
6900          GOSUB Generate_new
6910          GOSUB Gen_inverses
6940          One_in=Matrx(1,1,26)*(Xbar(6)-Xbar(8))^2+Matrx(2,2,26)*(Ybar(6)-Ybar(
8))^2
6950          One_in=One_in+2*Matrx(1,2,26)*(Xbar(6)-Xbar(8))*(Ybar(6)-Ybar(8))
6960          Two_in=Matrx(1,1,26)*(Xbar(6)-Xbar(9))^2+Matrx(2,2,26)*(Ybar(6)-Ybar(
9))^2
6970          Two_in=Two_in+2*Matrx(1,2,26)*(Xbar(6)-Xbar(9))*(Ybar(6)-Ybar(9))
6971          Test(4,5,7,Test2)
6980          !
6990          !
7000          PRINT TABXY(28,3);Isim
7010          IF Test2<-Prob_constant THEN
7020              ! Accept as same
7030              N_accept=N_accept+1
7040              PRINT TABXY(30,5);N_accept
7050              IF One_in>-Prob_constant THEN
7060                  Anotin1=Anotin1+1
7070                  PRINT TABXY(44,10);Anotin1
7080              END IF
7090              IF Two_in>-Prob_constant THEN
7100                  Anotin2=Anotin2+1
7110                  PRINT TABXY(44,11);Anotin2
7120              END IF
7130              IF One_in>-Prob_constant AND Two_in>-Prob_constant THEN
7140                  Anotinb=Anotinb+1
7150                  PRINT TABXY(44,12);Anotinb
7160              END IF
7170          ELSE
7180              N_reject=N_reject+1
7190              PRINT TABXY(30,6);N_reject
7200              IF One_in>-Prob_constant THEN
7210                  Rnotin1=Rnotin1+1
7220                  PRINT TABXY(44,15);Rnotin1

```

```

7230     END IF
7240     IF Two_in>-Prob_constant THEN
7250         Rnotin2=Rnotin2+1
7260         PRINT TABXY(44,16);Rnotin2
7270     END IF
7280     IF One_in>-Prob_constant AND Two_in>-Prob_constant THEN
7290         Rnotinb=Rnotinb+1
7300         PRINT TABXY(44,17);Rnotinb
7310     END IF
7320     END IF
7330     NEXT Isim
7340     PRINT TABXY(5,19);"*** SIMULATION COMPLETE ***";
7341     BEEP
7342     BEEP
7343     BEEP
7344     BEEP
7350     Pauseabit
7360     RETURN
7370     !
7380     ! Display results
7390     !
7400     Display_results:
7410     Clearscreen
7420     PRINTER IS 9;WIDTH 132
7421     PRINT
7422     PRINT "*****"
7423     PRINT
7424     PRINT
7430     PRINT "Results: ";
7440     IF Use_true=1 THEN
7450         PRINT "Using TRUE Covariance Matrices"
7460     ELSE
7470         PRINT "Using ESTIMATED Covariance Matrices"
7480     END IF
7490     PRINT
7491     FOR Imat=1 TO 2
7492         PRINT "Base Distribution #";Imat
7493         PRINT "Mean: (";Xbar(Imat);", ";Ybar(Imat);");"
7494         PRINT "Observations: ";Obs(Imat)
7495         PRINT "Covariance Matrix:"
7496         PRINT "  ";Matrx(1,1,Imat);"          ";Matrx(1,2,Imat);"  "
7497         PRINT "  ";Matrx(2,1,Imat);"          ";Matrx(2,2,Imat);"  "
7498         PRINT
7499     NEXT Imat
7500     FOR Imat=4 TO 5
7501         PRINT "Ellipse #";Imat-3
7502         PRINT "Mean: (";Xbar(Imat);", ";Ybar(Imat);");"
7504         PRINT "Covariance Matrix:"
7505         PRINT "  ";Matrx(1,1,Imat);"          ";Matrx(1,2,Imat);"  "
7506         PRINT "  ";Matrx(2,1,Imat);"          ";Matrx(2,2,Imat);"  "
7507         PRINT
7508     NEXT Imat
7590     PRINT "Last Combined Ellipse:"
7600     PRINT "Mean: (";Xbar(6);", ";Ybar(6);");"
7610     PRINT "Covariance Matrix:"
7620     PRINT "  ";Matrx(1,1,6);"          ";Matrx(1,2,6);"  "
7630     PRINT "  ";Matrx(2,1,6);"          ";Matrx(2,2,6);"  "
7640     PRINT
7650     PRINT "Simulation Results:"
7660     PRINT "# of simulations: ";Nsimulations
7661     PRINT
7662     PRINT "Totals;          % of Total;          % of Category"
7670     PRINT "Accepted:          ";N_accept;"          ";100*N_accept/Nsimulations
7680     PRINT "Target 1 NOT in: ";Anotin1;
7681     PRINT "          ";100*Anotin1/Nsimulations;"          ";100*Anotin1/N_accept
7690     PRINT "Target 2 NOT in: ";Anotin2;

```

```

7691 PRINT " ";100*Anotin2/Nsimulations;" ";100*Anotin2/N_accept
7700 PRINT " Neither one in: ";Anotinb;
7701 PRINT " ";100*Anotinb/Nsimulations;" ";100*Anotinb/N_accept
7710 PRINT "Rejected: ";N_reject;" ";100*N_reject/Nsimulations
7720 PRINT " Target 1 NOT in: ";Rnotin1;
7721 PRINT " ";100*Rnotin1/Nsimulations;" ";100*Rnotin1/N_reject
7730 PRINT " Target 2 NOT in: ";Rnotin2;
7731 PRINT " ";100*Rnotin2/Nsimulations;" ";100*Rnotin2/N_reject
7740 PRINT " Neither one in: ";Rnotinb
7741 PRINT " ";100*Rnotinb/Nsimulations;" ";100*Rnotinb/N_reject
7750 PRINT
7751 PRINT
7753 PRINT CHR$(12)
7760 PRINTER IS 1;WIDTH 80
7770 RETURN
7780 End_sub1:
7790 SUBEND
7800 SUB Power_ellipse
7810 ! Target Ellipse Chi-square Test Power Calculation
7820 ! Specify common variables
7830 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axi
a(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
7840 COM /Driver/ Obs(2)
7850 ! Variable Definition
7870 DIM New_mat(2,2,2),Delta(2)
7880 DIM Table(62,2)
7900 ! Initialize Program
7910 GOSUB Init_sub2
7920 ! MENU LOOP
7930 Menu_loop: !
7940 Clear_screen
7950 GOSUB Power_stuff
7960 SUBEXIT
7970 !
7980 ! INITIALIZE THE PROGRAM
7990 !
8000 Init_sub2: !
8010 Prob_constant=2*LOG(.05)
8020 ! Load in the Power Table
8030 RESTORE Power_data
8040 FOR I=1 TO 62
8050 FOR J=1 TO 2
8060 READ Table(I,J)
8070 NEXT J
8080 NEXT I
8170 RETURN
8180 !
8190 ! Power calculations: find the non-centrality parameter, Lambda.
8200 ! to use with the tables.
8210 !
8220 Power_stuff:
8230 PRINT "Power Calculations: Non-centrality Parameter"
8240 PRINT
8250 FOR I=1 TO 2
8260 FOR J=1 TO 2
8270 Power_mat(I,J,1)=Matrx(I,J,1)/Obs(1)+Matrx(I,J,2)/Obs(2)
8280 NEXT J
8290 NEXT I
8300 !compute inverse of power_mat
8310 Det=Power_mat(1,1,1)*Power_mat(2,2,1)-Power_mat(1,2,1)*Power_mat(1,2,1)
8320 Power_mat(1,1,2)=Power_mat(2,2,1)/Det
8330 Power_mat(2,2,2)=Power_mat(1,1,1)/Det
8340 Power_mat(1,2,2)=-Power_mat(1,2,1)/Det
8350 Power_mat(2,1,2)=-Power_mat(2,1,1)/Det
8360 PRINT "Enter the differences in the Mean components:"
8370 PRINT

```

```

8380 PRINT "Mean difference along the X axis: ";
8390 INPUT "X Difference:",Delta(1)
8400 PRINT Delta(1)
8410 PRINT "Mean difference along the Y axis: ";
8420 INPUT "Y Difference:",Delta(2)
8430 PRINT Delta(2)
8440 ! Calculate parameters
8450 Lambda=0
8460 FOR I=1 TO 2
8470     FOR J=1 TO 2
8480         Lambda=Lambda+Power_mat(I,J,2)*Delta(I)*Delta(J)
8490     NEXT J
8500 NEXT I
8510 PRINT "Lambda = ";Lambda
8520 PRINT
8530 ! Get Linear Interpolation Result
8540 GOSUB Linear_interp
8550 PRINT "Linear Power: ";Li_power
8560 ! Get Lagrange Interpolation Result
8570 GOSUB Lagrange_interp
8580 PRINT "Lagrange Power: ";La_power
8590 Pauseabit
8600 IF Delta(1)<>0 OR Delta(2)<>0 THEN 8360
8610 RETURN
8620 !
8630 ! Linear Interpolation
8640 !
8650 Linear_interp: !
8660 ! Find Bounding Values
8670 Search=1
8680 IF Lambda>=39 THEN
8690     Li_power=1
8700     RETURN
8710 END IF
8720 WHILE Lambda>Table(Search,1)
8730     Search=Search+1
8740 END WHILE
8750 IF Lambda=Table(Search,1) THEN
8760     Li_power=Table(Search,2)
8770     RETURN
8780 END IF
8790 Lambda_h=Table(Search,1)
8800 Lambda_l=Table(Search-1,1)
8810 Power_h=Table(Search,2)
8820 Power_l=Table(Search-1,2)
8830 IF Power_h=Power_l THEN
8840     Li_power=Power_h
8850     RETURN
8860 END IF
8870 Power1=Power_h*(Lambda-Lambda_l)/(Lambda_h-Lambda_l)
8880 Power2=Power_l*(Lambda_h-Lambda)/(Lambda_h-Lambda_l)
8890 Li_power=Power1+Power2
8900 RETURN
8910 ! Lagrange Interpolation
8920 Lagrange_interp: !
8930 Search=1
8940 IF Lambda>=39 THEN
8950     La_power=1
8960     RETURN
8970 END IF
8980 WHILE Lambda>Table(Search,1)
8990     Search=Search+1
9000 END WHILE
9010 IF Lambda=Table(Search,1) THEN
9020     La_power=Table(Search,2)
9030     RETURN

```

```

9040     END IF
9050     Summer=0
9060     FOR I=Search-3 TO Search+2
9070         Prod=Table(I,2)
9080         FOR J=Search-3 TO Search+2
9090             IF J<>I THEN
9100                 Prod=Prod*(Lambda-Table(J,1))/(Table(I,1)-Table(J,1))
9110             END IF
9120         NEXT J
9130         Summer=Summer+Prod
9140     NEXT I
9150     La_power=Summer
9160 RETURN
9170 !Chi-square(2) Power Table. 1st column is non-centrality parameter,
9180 !2nd is power. 0.05 significance label. From Selected Tables in
9190 !Mathematical Statistics, Volume 1.
9200 Power data:  !
9210 DATA .0,.05
9220 DATA .1,.0576
9230 DATA .2,.0653
9240 DATA .3,.0733
9250 DATA .4,.0814
9260 DATA .5,.0896
9270 DATA .6,.0980
9280 DATA .7,.1065
9290 DATA .8,.1151
9300 DATA .9,.1239
9310 DATA 1,.1327
9320 DATA 1.2,.1507
9330 DATA 1.4,.1691
9340 DATA 1.6,.1877
9350 DATA 1.8,.2065
9360 DATA 2.0,.2255
9370 DATA 2.2,.2447
9380 DATA 2.4,.2639
9390 DATA 2.6,.2831
9400 DATA 2.8,.3024
9410 DATA 3.0,.3215
9420 DATA 3.5,.3690
9430 DATA 4,.4154
9440 DATA 4.5,.4604
9450 DATA 5,.5037
9460 DATA 6,.5840
9470 DATA 7,.6554
9480 DATA 8,.7176
9490 DATA 9,.7707
9500 DATA 10,.8154
9510 DATA 11,.8526
9520 DATA 12,.8832
9530 DATA 13,.9080
9540 DATA 14,.9280
9550 DATA 15,.9440
9560 DATA 16,.9567
9570 DATA 17,.9667
9580 DATA 18,.9745
9590 DATA 19,.9805
9600 DATA 20,.9852
9610 DATA 21,.9888
9620 DATA 22,.9916
9630 DATA 23,.9937
9640 DATA 24,.9953
9650 DATA 25,.9965
9660 DATA 26,.9974
9670 DATA 27,.9981
9680 DATA 28,.9986
9690 DATA 29,.9989

```

9700 DATA 30,.9992
9710 DATA 31,.9994
9720 DATA 32,.9996
9730 DATA 33,.9997
9740 DATA 34,.9998
9750 DATA 35,.9998
9760 DATA 36,.9999
9770 DATA 37,.9999
9780 DATA 38,.9999
9790 DATA 39,1.0000
9800 DATA 40,1.0000
9810 DATA 41,1.0000
9820 DATA 42,1.0000
9830 End sub2: 1
9840 SUBEND

C. ELLIPSTUFF: Ellipse Routine Library

Ellipstuff is an "ellipse routine library," which contains a large number of routines to make working with ellipses easier. As such, both Genellipse and Ellsim contain all of these routines, but they are also listed separately. If more programs need to be written, these are the routines to build them with.

```

10  ! TEST PROGRAM FOR ELLIPSE ROUTINE LIBRARY
11  ! LIBRARY SUBROUTINES BEGIN ON LINE 1000.
12  ! THIS TEST PROGRAM GETS AN ELLIPSE FROM THE USER AND DISPLAYS IT ON
13  ! THE SCREEN.
20  GCLEAR
30  GRAPHICS ON
31  REAL Prob_constant
32  Prob_constant=2*LOG(.05)
40  Get_ell_mean(1)
50  ! Get_covariance(1)
60  Get_axes(1)
70  Invert(1,21)
80  Get_bounds(1,Prob_constant)
81  Choose_plotter
90  Draw_ellipse(1,-50,50,-50,50,Prob_constant)
100 STOP
110 END

1000 ! ELLIPSE ROUTINE LIBRARY....
1010 ! This file contains standard ellipse routines and data variables,
1020 ! including the following:
1030 !   -- Entry of ellipses by covariance matrices.
1040 !   -- Entry of ellipses by axes and orientation.
1050 !   -- Display of ellipses on screen and plotter.
1060 !   -- 2x2 Matrix inversion routine for use with Matrx.
1070 !   -- Matrx, an array which stores covariance matrices and their
1080 !       inverses (up to 10 matrices).
1090 !   -- Axes/orientation to Covariance matrix conversion routine.
1100 !
1110 ! USING MATRX
1120 ! Matrx is designed to hold covariance matrices and their inverses
1130 ! for 10 ellipses. In general, Matrx(1)...Matrx(20) are the covariance
1140 ! matrices and Matrx(21)...Matrx(40) are the corresponding inverses.
1150 ! This is the convention assumed by a number of these routines.
1160 ! In cases where the inverse is calculate first, put in Matrx(25), say,
1170 ! and then call Invert(25, 5). This will put the covariance matrix in
1180 ! Matrx(5).
1190 !
1200 ! THE ROUTINES ARE CALLED AS FOLLOWS:
1210 ! Invert(Srce, Dest)
1220 !   This command will invert the covariance matrix in Matrx(Srce) and
1230 !   put the result in Matrx(Dest)
1240 ! Get_Bounds(Ellipse, Prob_constant)
1250 !   This command will get the extreme points of the ellipse. Note that
1260 !   the ellipse's covariance matrix must have been inverted.
1270 ! Draw_Ellipse(Ellipse, Xmin, Xmax, Ymin, Ymax, Prob_constant)
1280 !   This will draw the given ellipse. The remaining variables define
1290 !   the screen. Note that Get_bounds must have been executed.
1300 ! Get_Ell_Mean(Ellipse)
1310 !   This is an input routine to read in Xbar and Ybar for the given
1320 !   ellipse.
1330 ! Get_Covariance(Ellipse)
1340 !   This is an input routine to read in the covariance matrix for the
1350 !   given ellipse.
1360 ! Get_Axes(Ellipse)
1370 !   This is an input routine which reads in the axes and orientation
1380 !   of the ellipse, and converts this to covariance matrix form.
1390 ! Choose_Plotter
1400 !   This prompts the user to draw the ellipse on the plotter or the
1410 !   screen.
1420 ! Display_Cov(Ellipse)
1430 !   This command displays the given covariance matrix.
1431 ! Test(Ellipse1, Ellipse2, Work, Test_Stat)
1432 !   This calculates the acceptance test criteria for the two ellipses.
1433 !   Work is a Matrx entry used as working space.
1434 ! Add_Covariance(Ellipse1, Ellipse2, Summar)

```

```

1435 ! This adds any two Matrix entries into a third Matrix entry.
1436 ! Combine_Ellipse(Ellipse1,Ellipse2,Combo)
1437 ! This combines any two Matrix entries into a third Matrix entry.
1438 ! Disp_Extremes(Ellipse)
1439 ! This displays the extreme X and Y bounds for the ellipse.
1441 ! Also included are Pauseabit and Clearscreen.
1450 !
1460 !-----!
1470 SUB Invert(Srce, Dest)
1480 ! This routine inverts any covariance matrix in Matrix and places
1490 ! the inverted matrix in Dest.
1500 COM /Ellipses/ Matrix(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
1510 is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
1520 Det=Matrix(1,1,Srce)*Matrix(2,2,Srce)-Matrix(1,2,Srce)*Matrix(1,2,Srce)
1530 Matrix(1,1, Dest)=Matrix(2,2,Srce)/Det
1540 Matrix(2,2, Dest)=Matrix(1,1,Srce)/Det
1550 Matrix(1,2, Dest)=-Matrix(1,2,Srce)/Det
1560 Matrix(2,1, Dest)=-Matrix(2,1,Srce)/Det
1560 SUBEND ! End of SUB Invert
1600 SUB Get_bounds(Ellipse, Prob_constant)
1610 ! This subroutine calculates the X and Y limits for the given ellipse
1620 COM /Ellipses/ Matrix(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
1630 is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
1640 REAL Temp1,Temp2,Temp3,Temp4,Temp5
1650 Temp1=Matrix(2,2,Ellipse+20)*Prob_constant
1660 Temp2=Matrix(1,2,Ellipse+20)*Matrix(1,2,Ellipse+20)-Matrix(1,1,Ellipse+20
1670 )*Matrix(2,2,Ellipse+20)
1680 Temp3=(Temp1/Temp2)^(.5)
1690 Temp4=Matrix(1,1,Ellipse+20)*Prob_constant
1700 Temp5=(Temp4/Temp2)^(.5)
1710 Emin_y(Ellipse)=-Temp5*Ybar(Ellipse)
1720 Emax_y(Ellipse)=Temp5*Ybar(Ellipse)
1730 Emin_x(Ellipse)=-Temp3*Xbar(Ellipse)
1740 Emax_x(Ellipse)=Temp3*Xbar(Ellipse)
1740 SUBEND ! End of GET_BOUNDS
1780 SUB Choose_plotter
1790 ! Subroutine to choose the desired plotter device (also contained
1791 ! in GRAPHER).
1800 REPEAT
1810 Go on=1
1820 DISP "Do you want to (D)raw, (E)rase, Plot with Pen (0), (1), (2)?"
1830 LINPUT Message$
1840 SELECT Message$
1850 CASE "D","d"
1860 PLOTTER IS 3,"INTERNAL"
1870 GRAPHICS ON
1880 PEN 1
1890 Go on=1
1891 CASE "0"
1892 PLOTTER IS 705,"HPGL"
1893 GRAPHICS ON
1894 PEN 0
1895 PLOTTER IS 3,"INTERNAL"
1896 GRAPHICS ON
1898 PEN 1
1899 Go on=1
1900 CASE "1"
1901 PLOTTER IS 705,"HPGL"
1902 GRAPHICS ON
1903 PEN 1
1904 Go on=1
1905 CASE "2"
1906 PLOTTER IS 705,"HPGL"
1907 GRAPHICS ON
1908 PEN 1

```

```

1944         PEN 2
1945         Go on=1
1950         CASE "E","e"
1960             PLOTTER IS 3,"INTERNAL"
1970             GRAPHICS ON
1980             PEN -1
1990             Go on=1
2000         CASE ELSE
2010             PRINT CHR$(7)
2020         END SELECT
2030     UNTIL Go_on=1
2040 SUBEND
2080 SUB Get_ell_mean(Ellipse)
2090     ! This subroutine prompts the user for the mean of an ellipse.
2100     COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
2110     Clearscreen
2120     PRINT TABXY(1,10);"WHAT IS X-BAR for ELLIPSE ";Ellipse;"?";
2130     INPUT Xbar(Ellipse)
2140     PRINT Xbar(Ellipse)
2150     PRINT TABXY(1,11);"WHAT IS Y-BAR for ELLIPSE ";Ellipse;"?";
2160     INPUT Ybar(Ellipse)
2170     PRINT Ybar(Ellipse)
2180     Pauseabit
2190 SUBEND
2230 SUB Get_covariance(Ellipse)
2240     ! This routine gets the covariance matrix for an ellipse
2250     COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
2260     Clearscreen
2270     FOR K=1 TO 2
2280         PRINT TABXY(1,3+K*6);"ENTER ELEMENT (";K;" ";K;" ) IN THE COVARIANCE
MATRIX FOR ELLIPSE#";Ellipse;" : ";
2290         INPUT Matrx(K,K,Ellipse)
2300         PRINT Matrx(K,K,Ellipse)
2310     NEXT K
2320     PRINT TABXY(1,3+9);"ENTER ELEMENT (1,2) IN THE COVARIANCE MATRIX FOR E
LEMENT#";I;" : ";
2330     INPUT Matrx(1,2,Ellipse)
2340     Matrx(2,1,Ellipse)=Matrx(1,2,Ellipse)
2350     PRINT Matrx(1,2,Ellipse)
2360     Pauseabit
2370 SUBEND
2410 SUB Pauseabit
2420     ! Pause and wait for a carriage return
2430     DISP "Type ENTER to continue...";
2440     INPUT Garbage$
2450 SUBEND
2460 SUB Clearscreen
2470     ! Clear the screen
2480     PRINT CHR$(12)
2490 SUBEND
2530 SUB Get_axes(Ellipse)
2540     ! This subroutine gets an ellipse in terms of the axes and the
2550     ! angle of orientation. These are converted into a covariance
2560     ! matrix.
2570     COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
2580     Prob_constant=2*LOG(.05)
2590     Clearscreen
2600     PRINT "WHAT IS THE RADIUS OF THE MAJOR AXIS OF ELLIPSE ";Ellipse;" : "
;
2610     INPUT Major_axis(Ellipse)
2620     PRINT Major_axis(Ellipse)
2630     PRINT "WHAT IS THE RADIUS OF THE MINOR AXIS OF ELLIPSE ";Ellipse;" : "
;

```

```

2640     INPUT Minor_axis(Ellipse)
2650     PRINT Minor_axis(Ellipse)
2660     PRINT "WHAT IS THE ANGLE (IN DEGREES) OF THE MAJOR AXIS OF ELLIPSE ";E
1lipse;" : ";
2670     INPUT Angle(Ellipse)
2680     PRINT Angle(Ellipse)
2690     ! Convert to Covariance Matrix
2700     Theta(Ellipse)=PI*Angle(Ellipse)/180
2710     Matrx(1,1,Ellipse)=(Major_axis(Ellipse)*COS(Theta(Ellipse)))^2+(Minor_
axis(Ellipse)*SIN(Theta(Ellipse)))^2
2720     Matrx(2,2,Ellipse)=(Major_axis(Ellipse)*SIN(Theta(Ellipse)))^2+(Minor_
axis(Ellipse)*COS(Theta(Ellipse)))^2
2730     Matrx(2,1,Ellipse)=((Major_axis(Ellipse))^2-(Minor_axis(Ellipse))^2)*C
OS(Theta(Ellipse))*SIN(Theta(Ellipse))
2740     Matrx(1,2,Ellipse)=Matrx(2,1,Ellipse)
2750     FOR J=1 TO 2
2760         FOR K=1 TO 2
2770             Matrx(J,K,Ellipse)=Matrx(J,K,Ellipse)/(-Prob_constant)
2780         NEXT K
2790     NEXT J
2800     PRINT
2810     Pauseabit
2820 SUBEND
2860 SUB Display_cov(Ellipse)
2870     ! Displays the covariance matrix for an ellipse
2880     COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
2890     FOR I=1 TO 2
2900         FOR J=1 TO 2
2910             PRINT Matrx(I,J,Ellipse);" ";
2920         NEXT J
2930     PRINT
2940     NEXT I
2950 SUBEND
2990 SUB Add_covariance(First,Second,Summer)
3000     ! This subroutine can be used to add Summer=First+Second
3010     COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
3020     INTEGER I,J
3030     FOR I=1 TO 2
3040         FOR J=1 TO 2
3050             Matrx(I,J,Summer)=Matrx(I,J,First)+Matrx(I,J,Second)
3060         NEXT J
3070     NEXT I
3080 SUBEND
3120 SUB Combine_ellipse(First,Second,Combo)
3130     ! This routine finds the "JPL" combination of the First and Second
3140     ! ellipses.
3150     COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
3160     REAL Sx1,Sx2
3170     ! Calculate the new covariance matrix.
3180     Invert(First,First+20)
3190     Invert(Second,Second+20)
3200     Add_covariance(First+20,Second+20,Combo+20)
3210     Invert(Combo+20,Combo)
3220     ! Calculate the new mean point.
3230     Sx1=Matrx(1,1,First+20)*Xbar(First)+Matrx(1,2,First+20)*Ybar(First)
3240     Sx1=Sx1+Matrx(1,1,Second+20)*Xbar(Second)+Matrx(1,2,Second+20)*Ybar(Se
cond)
3250     Sx2=Matrx(2,1,First+20)*Xbar(First)+Matrx(2,2,First+20)*Ybar(First)
3260     Sx2=Sx2+Matrx(2,1,Second+20)*Xbar(Second)+Matrx(2,2,Second+20)*Ybar(Se
cond)
3270     Xbar(Combo)=Matrx(1,1,Combo)*Sx1+Matrx(1,2,Combo)*Sx2
3280     Ybar(Combo)=Matrx(2,1,Combo)*Sx1+Matrx(2,2,Combo)*Sx2
3290 SUBEND

```

```

3330 SUB Disp_extremes(Ellipse)
3340 ! This routine displays the extreme x and y values for the specified
3350 ! ellipse
3360 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
3370 PRINT "E# ";Ellipse;" ";
3380 PRINT "Xmin ";Emin_x(Ellipse);
3390 PRINT " "; Xmax ";Emax_x(Ellipse);
3400 PRINT " "; Ymin ";Emin_y(Ellipse);
3410 PRINT " "; Ymax ";Emax_y(Ellipse)
3420 SUBEND
3430 SUB Draw_ellipse(Ellipse,Xmin,Xmax,Ymin,Ymax,Prob_constant)
3440 ! This routine draws one (1) ellipse on the current plotter device.
3450 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
3460 INTEGER Sign
3470 REAL Xpoint,Upoint,Ypoint,Temp1,Temp2,Temp3,Temp4,Temp5
3480 Get_bounds(Ellipse)
3490 Invert(Ellipse,Ellipse+20)
3500 GRAPHICS ON
3510 ! Draw top half of the ellipse
3520 Sign=1
3530 GOSUB Draw_half
3540 ! Draw bottom half of the ellipse
3550 Sign=-1
3560 GOSUB Draw_half
3570 ! Finish up
3580 MOVE (Xbar(Ellipse)-Xmin)*100/(Xmax-Xmin),(Ybar(Ellipse)-Ymin)*100/(Ym
ax-Ymin)
3590 DRAW (Xbar(Ellipse)-Xmin)*100/(Xmax-Xmin),(Ybar(Ellipse)-Ymin)*100/(Ym
ax-Ymin)
3600 ! Okay, dokey, we're done here.
3610 SUBEXIT
3620 Draw_half: ! Draw half of the ellipse
3630 ! If Sign=1, then draw top half; if Sign=-1, then bottom half.
3640 FOR Xpoint=Emin_x(Ellipse) TO Emax_x(Ellipse) STEP .1
3650 GOSUB Draw_1_point
3660 NEXT Xpoint
3670 Xpoint=Emax_x(Ellipse)
3680 GOSUB Draw_1_point
3690 RETURN
3700 !
3710 ! Compute each point and draw the new line. (It's here since we call
3720 ! it twice
3730 !
3740 Draw_1_point: !
3750 Upoint=Xpoint-Xbar(Ellipse)
3760 Temp1=Matrx(1,2,Ellipse+20)*Upoint
3770 Temp2=Temp1*Temp1-Matrx(2,2,Ellipse+20)*(Matrx(1,1,Ellipse+20)*Upoint*
Upoint+Prob_constant)
3780 IF Temp2<10*(-10) THEN Temp2=0
3790 Ypoint=(-Temp1+Sign*SQR(Temp2))/Matrx(2,2,Ellipse+20)+Ybar(Ellipse)
3800 IF Xpoint<=Emin_x(Ellipse)+.001 THEN
3810 MOVE (Xpoint-Xmin)*100/(Xmax-Xmin),(Ypoint-Ymin)*100/(Ymax-Ymin)
3820 ELSE
3830 DRAW (Xpoint-Xmin)*100/(Xmax-Xmin),(Ypoint-Ymin)*100/(Ymax-Ymin)
3840 END IF
3850 RETURN
3860 SUBEND
3900 SUB Test(First,Second,Work,Test_val)
3910 ! This routines calculates the acceptance test criteria for First and
3920 ! Second. Work is used as a "scratchpad".
3930 COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_ax
is(20),Theta(20),Angle(20),Emin_x(20),Emax_x(20),Emin_y(20),Emax_y(20)
3940 Add_covariance(First,Second,Work)
3950 Invert(Work,Work+20)

```

```
3960      Diffx=Xbar(First)-Xbar(Second)
3970      Diffy=Ybar(First)-Ybar(Second)
3980      Temp1=Matrx(1,1,Work+20)*Diffx+Matrx(1,2,Work+20)*Diffy
3990      Temp2=Matrx(2,1,Work+20)*Diffx+Matrx(2,2,Work+20)*Diffy
4000      Test_val=Diffx*Temp1+Diffy*Temp2
4010      SUBEND
```

END

Dtic

5-86