Confidence Ellipse Research Software

EAAF

Technical Memorandum No. 6

August 8, 1985

by

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

Concur:

James W. Grills, 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
**CONFIDENCE ELLIPSE RESEARCH SOFTWARE**

Dr. Janet Myhre, Will Duquette, Daniel Hockman

Mathematical Analysis Research Corp.  
4239 Via Padova  
Claremont, CA 91711

Commander, USAICS  
ATTN: ATSI-CD-SF  
Ft. Huachuca, AZ 85613-7000

Jet Propulsion Laboratory  
California Institute of Technology  
4800 Oak Grove Drive, ATTN: 126-200  
Pasadena, CA 91109

Approved for Public Distribution

Prepared under contract to JPL for the U.S. Army Intelligence Center and School's Combat Developer's Support Facility

CONFIDENCE ELLIPSE, ELLIPTICAL ERROR PROBABLE (EEP), COVARIANCE MATRIX, BASIC, STATISTICAL TESTS, MATHEMATICAL SIMULATION

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.
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
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

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 GENELLIPESE: 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 option 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).
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 \( \hat{S}^2 \) 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.
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(Ellipsel, Ellipse2, Combo)
This command combines the ellipses and point estimates in slots
Ellipsel 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 \(-\frac{2\ln(1-\alpha)}{\alpha}\), where \(\alpha\)
is the confidence level. For some reason, all of the programming
was done in terms of \(-\frac{2\ln(1-\alpha)}{\alpha}\). 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(1)

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.
This program uses the general ellipse generation and graphing routines found in the file ELLIPSTUFF. Its purpose is to graph general ellipses and combinations of ellipses.

Will Duquette
May 13, 1985

REAL Xfin,Xmax,Ymin,Ymax,Prob_constant

Init prog:

These variables define the size of the screen. For best results, the lengths at the X and Y axes should be about the same.

Prob_constant=2*LOG(.1)

Display the menu

Menu top:

Clear screen

PRINT "GENERAL ELLIPSE GRAPHER"

PRINT "--------------------------"

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

PRINT "Enter your choice:"

DISP "Choose an option:"
INPUT Option$ SELECT Option$ CASE "A","a" ! Get ellipse mean points.

Clear screen

PRINT "Getting mean point for Ellipse#;"
REPEAT
Ellipse=0
DISP "Enter the Ellipse# (1-20)";
INPUT Ellipse
PRINT Ellipse
IF Ellipse<20 AND Ellipse>0 THEN CALL Get_0llmeanCEllIps*)
UNTIL Ellipse=0
CASE "B","b" ! Get ellipse covariance matrices
Clear screen

PRINT "Getting Shape (Covariance Matrix) for Ellipse#;"
REPEAT
Ellipse=0
DISP "Enter the Ellipse# (1-20)";
INPUT Ellipse
PRINT Ellipse
IF Ellipse<20 AND Ellipse>0 THEN CALL Get_covariance(Ellipse)
UNTIL Ellipse=0
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 Ellipse3<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 "f","F" ! Graph ellipses
1061 REPEAT
1070 CLEARSCREEN
1080 PRINT "Graphing Ellipse#";
1090 Ellipse=0
1100 DISP "Enter Ellipse# (1-20)";
1120 INPUT Ellipse
1130 PRINT Ellipse
1140 IF Ellipse<20 AND Ellipse>0 THEN
1150 PRINT
1160 PRINT "Center Point (",Xbar(Ellipse),",",Ybar(Ellipse),")"
1170 PRINT
1180 PRINT "Covariance Matrix:";
1190 Display_cov(Ellipse)
1200 PRINT "Major: ",Major_axis(Ellipse)
1210 PRINT "Minor: ",Minor_axis(Ellipse)
1220 PRINT "Orient: ",Angle(Ellipse)
1230 Invert(Ellipse,Ellipse+20)
1240 Get_bounds(Ellipse,Prob_constant)
1250 Draw_ellipse(Ellipse,Xmin,Xmax,Ymin,Ymax,Prob_constant)
1260 ALPHA ON
1270 END IF
1280 UNTIL Ellipse=0
1290 CASE "g","G" ! Clear the graphics screen
1300 GCLEAR
1301 CASE "H","h" ! Choose the plotter device
1302 Choose_plotter
1310 CASE "k","K" ! Set the screen boundaries
1320 CLEARSCREEN

A-3
PRINT "Setting Screen Boundaries...."
PRINT
REPEAT
  Ellipse=0
  DISP "Display Extreme Points for Ellipse# (1-20)"
  INPUT Ellipse
  IF Ellipse>0 AND Ellipse<=20 THEN
    Invert(Ellipse,Ellipse+20)
    Get_bounds(Ellipse,Prob_constant)
    Disp_extreme(Ellipse)
  END IF
UNTIL Ellipse=0

PRINT

Get_x:
PRINT "Minimum X value: ";
INPUT Xmin
PRINT Xmin
PRINT "Maximum X value: ";
INPUT Xmax
PRINT Xmax
IF Xmin>Xmax THEN Get_x

Get_y:
PRINT "Minimum Y value: ";
INPUT Ymin
PRINT Ymin
PRINT "Maximum Y value: ";
INPUT Ymax
PRINT Ymax
IF Ymin>Ymax THEN Get_y
CASE "I","I"
  Calculate the chi-square acceptance test statistic
  Clearscreen
  PRINT "Calculating Acceptance Test"
  PRINT
  PRINT "Ellipse 1 #";
  DISP "Enter the first Ellipse# (1-20)"
  INPUT Ellipse1
  PRINT Ellipse1
  PRINT "Ellipse 2 #";
  DISP "Enter the second Ellipse# (1-20)"
  INPUT Ellipse2
  PRINT Ellipse2
  PRINT "Scratch Work #";
  DISP "Enter the scratch work # (1-20)"
  INPUT Swork
  PRINT Swork
  IF Ellipse1<20 AND Ellipse1>0 AND Ellipse2<20 AND Ellipse2>0 AND Swork<20 AND Swork>0 AND Swork>Ellipse1 AND Swork>Ellipse2 THEN
    Test(Ellipse1,Ellipse2,Swork,Test_stat)
  END IF
  PRINT
  PRINT "The Test statistic is ";Test_stat
  PRINT
  PAUSE wait
CASE "J","J"
  Draw in the axes
  GRAPHICS ON
  CSIZE 2
  MOVE 0,0
  DRAW 0,100
1927 MOVE 100.0
1928 DRAW 0.0
1929 LABEL \(";Xmin:"",";Ymin:"\)
1930 LONG 7
1931 MOVE 100.0
1932 LABEL Xmax
1933 LONG 3
1934 MOVE 0.100
1935 LABEL Ymax
1936 LONG 3
1937 MOVE 50.0
1938 CSIZE 3
1939 MOVE \(\text{"X-Axis (in kilometers)"}\)
1940 LABEL Xmaz
1941 LORG 7
1942 MOVE 0.100
1943 LABEL Ymax
1944 LORG 3
1945 LONG 1
1946 LDIR 0
1947 RAD
1948 CASE \("x","x"\) ! We can stop now
1949 CLEARSCREEN
1950 PRINT \"That\'s all, folks!\"
1951 STOP
1952 CASE ELSE
1953 PRINT CHRS(7)
1954 END SELECT
1955 GOTO Menu_top
1956 END
1957 SUB SUBROUTINES: Taken from ELLIPSTUFF
1958 SUB Invert(Source,Dest)
1959 ! This routine inverts any covariance matrix in Source and places
1960 ! the inverted matrix in Dest.
1961 MATRX(MATRX(2,2,Source),Xbar(20),Xbar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Emin_x(20),Emin_y(20),Emax_x(20),Emax_y(20))
1962 Det=MATRX(1,1,Source)*MATRX(2,2,Source)-MATRX(1,2,Source)*MATRX(2,1,Source)
1963 MATRX(1,1,Dest)=MATRX(2,2,Source)/Det
1964 MATRX(2,2,Dest)=MATRX(1,1,Source)/Det
1965 MATRX(1,2,Dest)=-MATRX(1,2,Source)/Det
1966 MATRX(2,1,Dest)=-MATRX(2,1,Source)/Det
1967 SUBEND ! End of SUB Invert
1968 SUB GET_BOUNDS
1969 ! This subroutine calculates the X and Y limits for the given ellipse
1970 CON /Elliipse/ MATRX(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Emin_x(20),Emin_y(20),Emax_x(20),Emax_y(20)
1971 REAL Temp1,Temp2,Temp3,Temp4,Temp5
1972 Temp1=MATRX(2,2,Ellipse)+20*Prob_constant
1973 Temp2=MATRX(1,1,Ellipse)+20*Prob_constant
1974 Temp3=(Temp1/Temp2)^.5
1975 Temp4=MATRX(1,1,Ellipse)+20*Prob_constant
1976 Temp5=(Temp4/Temp2)^.5
1977 Emin_y(Ellipse)=Temp5*Ybar(Ellipse)
1978 Emin_x(Ellipse)=Temp3*Xbar(Ellipse)
1979 Emax_y(Ellipse)=Temp5*Ybar(Ellipse)
1980 Emax_x(Ellipse)=Temp3*Xbar(Ellipse)
1981 SUBEND ! End of GET_BOUNDS
1982 ! CHOOSE PLOTTER
1983 !
SUB Choose_plotter
  ! Subroutine to choose the desired plotter device
  REPEAT
    Go on=1
    DISPLAY "Do you want to (D)raw, (E)rase, Plot with Pen (0), (1), (2)?"
    INPUT Message$
    SELECT Message$
      CASE "D","d"
        PLOTTER IS 3,"INTERNAL"
        GRAPHICS ON
        PEN 1
        Go on=1
      CASE "E","e"
        PLOTTER IS 705,"HPGL"
        GRAPHICS ON
        PEN 2
        Go on=1
      CASE "1"
        PLOTTER IS 705,"HPGL"
        GRAPHICS ON
        PEN 0
        Go on=1
      CASE ELSE
        PRINT CHR$(7)
        END SELECT
      UNTIL Go on=1
  SUBEND

SUB Get_ell_mean(Ellipse)
  ! This subroutine prompts the user for the mean of an ellipse.
  COM/Ellipse/ Matrix(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20,Emax_x(20),Emin_x(20),Emax_y(20),Emin_y(20)
  Clearscreen
  PRINT TABXY(1,10);"ENTER ELEMENT (K,K) IN THE COVARIANCE MATRIX FOR ELLIPSE ";Ellipses;"
  INPUT Xbar(Ellipse)
  PRINT Ybar(Ellipse)
  PRINT TABXY(1,11);"ENTER ELEMENT (K,K) IN THE COVARIANCE MATRIX FOR ELLIPSE ";Ellipses;"
  INPUT Ybar(Ellipse)
  PRINT "Pausebit"
SUBEND

SUB Get_covariance(Ellipse)
  ! This routine gets the covariance matrix for an ellipse
  COM/Ellipse/ Matrix(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20,Emax_x(20),Emin_x(20),Emax_y(20),Emin_y(20)
  Clearscreen
  FOR K=1 TO 2
    PRINT TABXY(1,3*K-6);"ENTER ELEMENT (";K;",";K;") IN THE COVARIANCE MATRIX FOR ELLIPSE ";Ellipses;"
    INPUT Matrix(K,K,Ellipses)
  NEXT K

A-6
5860 PRINT Matrix(K,K,Ellipse)
5870 NEXT K
5880 PRINT TABX(1,3*9):"ENTER ELEMENT (1,2) IN THE COVARIANCE MATRIX FOR E
LEMENT#1": "
5890 INPUT Matrix(1,2,Ellipse)
5900 Matrix(2,1,Ellipse)=Matrix(1,2,Ellipse)
5910 PRINT Matrix(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/ Matrix(2,2,10),Xbar(20),Ybar(20),Major_axis(20),Minor_axis
is(20),Theta(20),Angle(20),Emin_x(20),Emin_y(20),Emax_x(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 Matrix(1,1,Ellipse)=(Major_axis(Ellipse)*COS(Theta(Ellipse)))^2+(Minor_axis
(Ellipse)*COS(Theta(Ellipse)))^2
6280 Matrix(2,2,Ellipse)=(Major_axis(Ellipse)*COS(Theta(Ellipse)))^2+(Minor_axis
(Ellipse)*COS(Theta(Ellipse)))^2*C
6290 Matrix(2,1,Ellipse)=(Major_axis(Ellipse)*SIN(Theta(Ellipse))).^2-(Minor_axis(Ellipse)^2)*C
6300 Matrix(1,2,Ellipse)=Matrix(2,1,Ellipse)
6310 FOR J=1 TO 2
6320 FOR K=1 TO 2
6330 Matrix(J,K,Ellipse)=Matrix(J,K,Ellipse)/(-Prob_constant)
6340 NEXT J
6350 NEXT K
6360 PRINT
6370 Pauseabit
6380 SUBEND
6390 !
6400 ! DISPLAY_COV
6410 !
6420 SUB Display_cov(Ellipse)
6430 ! Displays the covariance matrix for an ellipse

A-7
FOR I=1 TO 2
FOR J=1 TO 2
PRINT Matrx(I,J,Ellipse);" ";
NEXT J
PRINT
NEXT I
SUBEND

FOR 1-1 TO 2
FOR J.1 TO 2
PRINT Matrx(I,J,Ellipse);" ";
NEXT J
PRINT
NEXT I
SUBEND

SUB Add_covariance(First,Second,Summer)
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)
FOR I=1 TO 2
FOR J=1 TO 2
Matrx(I,J,Summer)=Matrx(I,J,First)+Matrx(I,J,Second)
NEXT J
NEXT I
SUBEND

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)

INTEGER I,J
REAL Sx1,Sx2

SUB Combine_ellipse(First,Second,Combo)
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)
REAL Sx1,Sx2

SUB Disp_extremes(Ellipse)
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)

SUB Draw_ellipse(Ellipse, Xmin,Xmax,Ymin,Ymax,Prob_constant)
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)
REAL Xpoint, Upoint, Ypoint, Temp1, Temp2, Temp3, Temp4, Temp5

GRAPHICS ON

DRAW top half of the ellipse

SIGN = 1

GOSUB Draw_half

DRAW bottom half of the ellipse

SIGN = -1

GOSUB Draw_half

! Finish up

MOVE (Xbar(Ellipse) -Xmin)*100/(Xmax-Xmin), (Ybar(Ellipse) -Ymin)*100/(Ymax-Ymin)

DRAW (Xbar(Ellipse) -Xmin)*100/(Xmax-Xmin), (Ybar(Ellipse) -Ymin)*100/(Ymax-Ymin)

! Okey, dokey, we're done here.

SUBEXIT

DRAW_half:

! Draw half of the ellipse

IF SIGN = 1, then draw top half; if SIGN = -1, then bottom half.

FOR Xpoint=Emin_x(Ellipse) TO Emax_x(Ellipse) STEP .1

GOSUB Draw_I_point

NEXT Xpoint

RETURN

! Compute each point and draw the new line. (It's here since we call

SUB I_point twice

SUBEND

! TEST

SUB Test(First, Second, Work, Test_val)

! This routine calculates the acceptance test criteria for First and

Second. Work is used as a "scratchpad".

COM /Ellipses/ Matrix(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)

Add_covariance(First, Second, Work)

Invert(Work, Work+20)

Diff_x=Xbar(First)-Xbar(Second)

Diff_y=Ybar(First)-Ybar(Second)

Temp1=Matrix(1,1,Work+20)*Diff_x+Matrix(1,2,Work+20)*Diff_y

Temp2=Matrix(2,1,Work+20)*Diff_x+Matrix(2,2,Work+20)*Diff_y

Test_val=Diff_x*Temp1+Diff_y*Temp2

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.
This is a simple simulation program for the ellipse combination test. Only the Normal distribution is supported.

Original: 1/17/85  Updated: 7/29/85

---

Top_of_program:  

Get choice:  

Get the number of TRIALS (Main Menu Choice "B")

Get_obs:  

TOR IMAT=1 TO 2
PRINT "How many observations for ellipse #"; Imat; ": ";
INPUT Obs(Imat)
PRINT Obs(lmat)
END

SUB Invert(Srce,Dest)
This routine inverts any covariance matrix in Matrx and places
the inverted matrix in Dent.
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)
Det=Matrx(1,1,Srce) #Matrx(1,2,Srce) - Matrx(1,2,Srce) #Matrx(1,1,Srce)
Matrx(1,1,Dest)=Matrx(2,2,Srce)/Det
Matrx(2,2,Dest)=Matrx(1,1,Srce)/Det
Matrx(1,2,Dest)=Matrx(1,2,Srce)/Det
Matrx(2,1,Dest)=Matrx(2,1,Srce)/Det
SUBEND ! End of SUB Invert

SUB GET_BOUNDS
This subroutine calculates the X and Y limits for the given ellipse
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)
REAL Temp1,Temp2,Temp3,Temp4,Temp5
Temp1=Matrx(2,2,Ellipse,20)*Prob_constant
Temp2=Matrx(1,2,Ellipse,20)*Matrx(1,1,Ellipse,20)-Matrx(1,1,Ellipse,20)*Matrx(1,2,Ellipse,20)
Temp3=(Temp1/Temp2)^(.5)
Temp4=Matrx(1,1,Ellipse,20)*Prob_constant
Temp5=(Temp4/Temp2)^(.5)
Emin_y(Ellipse)=Temp5+Ybar(Ellipse)
Emax_y(Ellipse)=Temp5+Ybar(Ellipse)
Emin_x(Ellipse)=Temp3+Xbar(Ellipse)
Emax_x(Ellipse)=Temp3+Xbar(Ellipse)
SUBEND ! End of GET_BOUNDS

SUB Choose_plotter
Subroutine to choose the desired plotter device
Go on-1
DISP "Do you want to (D)raw, (E)rase, Plot with Pen (0), (1), (2)?" 
LINPUT Message$
SELECT Message$
CASE "D","d"
PLOTTER IS 3,"INTERNAL"
GRAPHICS ON
PEN 1
Go on-1
CASE "E","e"
PLOTTER IS 705,"HPGL"
GRAPHICS ON
PEN 0
Go on-1
CASE "P","p"
PLOTTER IS 705,"HPGL"
GRAPHICS ON
PEN 1
B-3
Go on-
CASE "Z"
PLOTTER IS 705, "HPGL"
GRAPHICS ON
PEN 2
Go on-
CASE "E", "E"
PLOTTER IS 3, "INTERNAL"
GRAPHICS ON
PEN -1
Go on-
CASE ELSE
PRINT CHR$(7)
END SELECT
UNTIL Go on-
SUBEND

1
GET ELL_MEAN
SUB
Got ell mean(Elllpse)

Clearscreen
PRINT TABXY(I,10);"WHAT IS X-BAR for ELLIPSE ";Ellipse,"?"
INPUT Xbar(Ellipse)
PRINT Xbar(Ellipse)
PRINT TABXY(I,11);"WHAT IS Y-BAR for ELLIPSE ";Ellipse,"?"
INPUT Ybar(Ellipse)
PRINT Ybar(Ellipse)
Pauseabit
SUBEND

1
GET_COVARIANCE
SUB
Get covariance(Ellipse)

Clearscreen
FOR K=1 TO 2
PRINT TABXY(I,10);"ENTER ELEMENT (";K;",:) IN THE COVARIANCE MATRIX FOR ELLIPSE#";Ellipse;": ";
INPUT Matrx(K,K,Ellipse)
PRINT Matrx(K,K,Ellipse)
NEXT K
PRINT TABXY(1.2,Ellipse)
Pauseabit
SUBEND

1
PAUSEABIT
SUB
Pause and wait for a carriage return
DISP "Type ENTER to continue..."
INPUT Garbage$
I GET_AXES
SUB Get_axes(Ellipse)
This subroutine gets an ellipse in terms of the axes and the angle of orientation. These are converted into a covariance matrix.
COM /Ellipses/ Matrix(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Emin_x(20),Emin_y(20),Emax_x(20),Emax_y(20)
Prob_constant=2*LOG(.05)
Clearscreen
PRINT "WHAT IS THE RADIUS OF THE MAJOR AXIS OF ELLIPSE ":Ellipse; ";
INPUT Major_axis(Ellipse)
PRINT Major_axis(Ellipse)
PRINT "WHAT IS THE RADIUS OF THE MINOR AXIS OF ELLIPSE ":Ellipse; ";
INPUT Minor_axis(Ellipse)
PRINT Minor_axis(Ellipse)
PRINT "WHAT IS THE ANGLE (IN DEGREES) OF THE MAJOR AXIS OF ELLIPSE ":Ellipse; " ";
INPUT Angle(Ellipse)
PRINT Angle(Ellipse)
Convert to Covariance Matrix
Theta(Ellipse)=PI*Angle(Ellipse)/180
Matrix(1,1,Ellipse)=(Major_axis(Ellipse)*COS(Theta(Ellipse)))*2+(Minor_axis(Ellipse)*SIN(Theta(Ellipse)))*2
Matrix(2,2,Ellipse)=(Minor_axis(Ellipse)*COS(Theta(Ellipse)))*2-(Minor_axis(Ellipse)*SIN(Theta(Ellipse)))*2*COS(Theta(Ellipse))
Matrix(1,2,Ellipse)=Matrix(2,1,Ellipse)
FOR J=1 TO 2
FOR K=1 TO 2
Matrix(J,K,Ellipse)=Matrix(J,K,Ellipse)/(-Prob_constant)
NEXT K
NEXT J
PRINT Pauseabit
SUBEND
DISPLAY
SUB Display_cov(Ellipse)
Display the covariance matrix for an ellipse
COM /Ellipses/ Matrix(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Emin_x(20),Emin_y(20),Emax_x(20),Emax_y(20)
FOR J=1 TO 2
FOR K=1 TO 2
PRINT Matrix(I,J,Ellipse); " ";
NEXT J
PRINT NEXT I
SUBEND
ADD
SUB Add_covariance(First,Second,Summer)
This subroutine can be used to add Summer=First+Second
COM /Ellipses/ Matrix(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Emin_x(20),Emin_y(20),Emax_x(20),Emax_y(20)
INTEGER I,J
FOR I=1 TO 2
FOR J=1 TO 2
NEXT J
NEXT I
SUBEND
SUB Combine_ellipse(First, Second, Combo)

This routine finds the "JPL" combination of the First and Second ellipses.

COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Emax_x(20),Emay_x(20),Emax_y(20),Emin_y(20)

REAL Sx1,Sx2

Calculate the new covariance matrix.

Invert(First,First+20)

Invert(Second,Second+20)

Add covariance(First,Second+20,Combo+20)

Calculate the new mean point.

Sx1=Matrx(1,1,First+20)*Xbar(First)+Matrx(1,2,First+20)*Ybar(First)

Sx2=Matrx(2,1,Second+20)*Xbar(Second)+Matrx(2,2,Second+20)*Ybar(Second)

Xbar(Combo)=Matrx(1,1,Combo)*Sx1+Matrx(1,2,Combo)*Sx2

Sx2

SUBEND

SUB Disp_extremes(Ellipse)

This routine displays the extreme x and y values for the specified ellipse.

COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Emax_x(20),Emax_y(20),Emin_x(20),Emin_y(20)

PRINT "Xmin ":Emax_x(Ellipse)

PRINT "Xmax ":Emax_x(Ellipse)

PRINT "Ymin ":Emin_y(Ellipse)

PRINT "Ymax ":Emin_y(Ellipse)

SUBEND

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

This routine draws one (1) ellipse on the current plotter device.

COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Emax_x(20),Emax_y(20),Emin_x(20),Emin_y(20)

INTEGER Sign

REAL Xpoint,Upoint,Ypoint,Temp1,Temp2,Temp3,Temp4,Temp5

GRAPHICS ON

Draw top half of the ellipse.

Sign=1

GOSUB Draw_half

Sign=-1

GOSUB Draw_half

FINISH UP

MOVE (Xbar(Ellipse)-Xmin)*100/(Xmax-Xmin),(Ybar(Ellipse)-Ymin)*100/(Ymax-Ymin)

DRAW (Xbar(Ellipse)-Xmin)*100/(Xmax-Xmin),(Ybar(Ellipse)-Ymin)*100/(Ymax-Ymin)

GOSUB Draw_half

IF Sign=1, then draw top half; if Sign=-1, then bottom half.

FOR Xpoint=Emax_x(Ellipse) TO Emax_x(Ellipse) STEP .1

GOSUB Draw_half

NEXT Xpoint

GOSUB Draw Half

SUBEXIT

SUB Draw_half: Draw half of the ellipse

IF Sign=1, then draw top half; if Sign=-1, then bottom half.

FOR Xpoint=Emax_x(Ellipse) TO Emax_x(Ellipse) STEP .1

GOSUB Draw Half

NEXT Xpoint

GOSUB Draw Ellipse
RETURN

Compute each point and draw the new line. (It's here since we call it twice)

Draw_1_point:

Temp=Temp1*Matrx(1,2,Ellipse)*Upoint
Upoint=Prob constant

IF Temp2<10^(-10) THEN Temp2=0

Ypoint=(-Temp1*Sign*SQR(Temp2))/Matrx(2,2,Ellipse)*Ybar(Ellipse)

ELSE

DRAW (Xpoint-Xbar(Ellipse))100/(Xmax-Xmin),(Ypoint-Ybar(Ellipse))*100/(Ymax-Ymin)

END IF

RETURN

SUB END

TEST:

SUB Test(First,Second,Work,Test_val)

This routine calculates the acceptance test criteria for First and Second. Work is used as a "scratchpad".

COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Emin_x(20),Emin_y(20),Emax_x(20),Emax_y(20)

Add_covariance(First,Second,Work)

Invert(Work,Work+20)

Diffx=Xbar(First)-Xbar(Second)

Diffy=Ybar(First)-Ybar(Second)

Temp1=Matrx(1,1,Work+20)*Diffx*Matrx(1,2,Work+20)*Diffy

Temp2=Matrx(2,1,Work+20)*Diffx*Matrx(2,2,Work+20)*Diffy

Test_val=Diffx*Temp1+Diffy*Temp2

SUB END

SUB Sim_ellipse

Ellipses Combination Simulation Program

1/16/85 Update 7/18/85

Specify the common variables

COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Emin_x(20),Emin_y(20),Emax_x(20),Emax_y(20)

COM /Driver/ Obs(2)

Variable Definitions

INTEGER Ellnum,Imat,Xmax,Ymax,Imin,Xmin

INTEGER Mobservations(2)

DIM Sx(2,10),Xs(5000),Ys(5000)

Use of MATRIX Array

MATRX 1 is an initial matrix

MATRX 2 is an initial matrix

MATRX 3 is the theoretical combination of 8 and 9

MATRX 4 is the estimate of 8

MATRX 5 is the estimate of 9

MATRX 6 is the combination of 4 and 5

MATRX 7 is the sum of 3 and 4 (for test)

Note that if the True Variance-Covariance is used instead of the estimate, this is equal to the sum of 8 and 9.

MATRX 8 is MATRX 1 divided by sample size

MATRX 9 is MATRX 2 divided by sample size

MATRX 21-29 are the inverses of 1-9

Initialization
1 Main Program Loop
1 Main Loop: 1
1 Clearscreen
1 PRINT "Ellipse Simulations"
1 PRINT
1 PRINT "Enter the Letter of your choice:"
1 PRINT "(A) Reset the Random Number Seed"
1 PRINT "(B) Simulate using true Covariance Matrices"
1 PRINT "(C) Simulate using estimated Covariance Matrices"
1 PRINT "(D) Display Results on the Printer"
1 PRINT "(X) Exit Program"
1 Get option: 1
1 INPUT "Enter Your Choice:“, MainChoice$ 1
1 SELECT MainChoice$ 1
1 CASE "A", "A" 1
1 GOSUB Get seed 1
1 CASE "B", "B" 1
1 Use true=1 1
1 GOSUB Simulate 1
1 CASE "C", "C" 1
1 Use true=0 1
1 GOSUB Simulate 1
1 CASE "D", "D" 1
1 GOSUB Display_results 1
1 CASE "X", "X" 1
1 GOTO End sub1 1
1 CASE ELSE 1
1 5000 PRINT CHR$(7) 1
1 5010 GOTO Get option 1
1 5020 END SELECT 1
1 5030 GOTO Main loop 1
1 Utility Subroutines 1
1 Initialize Program 1
1 GOTO Main_loop 1
1 !----------------------------- 1
1 5040 ! Utility Subroutines 1
1 5050 ! Utility Subroutines 1
1 5060 !----------------------------- 1
1 5070 ! Initialize Program 1
1 5080 ! Initialize Program 1
1 5090 ! Initialize Program 1
1 5100 Init sub1: 1
1 5110 RANDOMIZE 1
1 5120 Prob_constant=2*LOG(.05) 1 Confidence level parameter 1
1 5130 FOR I=1 TO 2 1
1 5140 FOR J=1 TO 2 1
1 5150 Matrix(I,J,8)=Matrix(I,J,1)/Obs(1) 1
1 5160 Matrix(I,J,9)=Matrix(I,J,2)/Obs(2) 1
1 5170 NEXT J 1
1 5180 NEXT I 1
1 5190 Xbar(8)=Xbar(1) 1
1 5200 Ybar(8)=Ybar(1) 1
1 5210 Xbar(9)=Xbar(2) 1
1 5220 Ybar(9)=Ybar(2) 1
1 5230 ! Set parameters 1
1 5240 RETURN 1
1 5300 ! Generate the new ellipses using random observations 1
1 5350 ! Generate new: ****FLAG**** 1
1 5360 ! Generate new: ****FLAG**** 1
1 5370 FOR Imat=1 TO 2 1
1 5380 M=Obs(Imat) 1
1 5400 ! Generate the X and Y values 1
1 5410 Xaus=0 1
1 5420 Yaus=0 1
1 5430 Co=SQR(Matrix(I,1,Imat))
A-Sigma(1,2,imat)/Cc
Bee=SQRT(Matrix(2,2,imat)-A*A)
FOR I=1 TO M
U1=RND
U2=RND
X1=SQRT(-2*LOG(U1))*SIN(2*PI*U2)
Y1=SQRT(-2*LOG(U2))*COS(2*PI*U1)
Xs(I)=Cc*X1*Xbar(imat)
Ys(I)=A*X1*BeeOY1*Xbar(imat)
Xs(I)=Xs(I)+X(I)
Ys(I)=Ys(I)+Y(I)
NEXT I
Xbar(imat+3)=Xsum/M
Ybar(imat+3)=Ysum/M
Calculate the variances
Matrix(1,1,imat+3)=0
Matrix(2,2,imat+3)=0
Matrix(1,2,imat+3)=0
FOR I=1 TO M
Matrix(1,1,imat+3)=Matrix(1,1,imat+3)+(Xs(I)-Xbar(imat+3))*(Xs(I)-Xbar(imat+3))
Matrix(2,2,imat+3)=Matrix(2,2,imat+3)+(Ys(I)-Ybar(imat+3))*(Ys(I)-Ybar(imat+3))
Matrix(1,2,imat+3)=Matrix(1,2,imat+3)+(Xs(I)-Xbar(imat+3))*(Ys(I)-Ybar(imat+3))
NEXT I
Matrix(1,1,imat+3)=Matrix(1,1,imat+3)/(M-1)*M
Matrix(2,2,imat+3)=Matrix(2,2,imat+3)/(M-1)*M
Matrix(1,2,imat+3)=Matrix(1,2,imat+3)/(M-1)*M
NEXT Imat
RETURN ! Generate_new
Generate the inverses of our matrices
IF Use true=0 THEN Patch2
! To use the true matrices, copy them from 8 and 9 into 4 and 5
FOR Imat=8 TO 9
FOR J=1 TO 2
Matrix(I,J,imat-4)=Matrix(I,J,imat)
NEXT J
NEXT Imat
Patch2: 1
RETURN ! Gen_inverses
-------------------------------------------------------------------------------
Program Subroutines
-------------------------------------------------------------------------------
! GET SEED (Menu Menu Choice "C")
...
END IF
IF Two in>-Prob_constant THEN
    RnotIn2=RnotIn2+1
    PRINT TABXY(44,16);RnotIn2
END IF
IF One in>-Prob_constant AND Two in>-Prob_constant THEN
    RnotInb=RnotInb+1
    PRINT TABXY(44,17);RnotInb
END IF
NEXT Isim
PRINT TABXY(5,19);"*** SIMULATION COMPLETE ***";
BEEP
BEEP
BEEP
Pauseabit
RETURN

1
Display results
Clearscreen
PRINTER IS 9;WIDTH 132
PRINT "**************************************************************
PRINT "Results: ";
IF Use_true=1 THEN
PRINT "Using TRUE Covariance Matrices"
ELSE
PRINT "Using ESTIMATED Covariance Matrices"
END IF
FOR Imat=1 TO 2
    PRINT "Base Distribution #":Imat
    PRINT "Mean: (";Xbar(Imat);", ";Xbar(Imat);")
    PRINT "Observations: ";Obs(Imat)
    PRINT "Covariance Matrix:
    PRINT ";Matrx(1,1,Imat);" ;Matrx(1,2,Imat);" *
    PRINT ";Matrx(2,1,Imat);" ;Matrx(2,2,Imat);" *
PRINT NEXT Imat
FOR Imat=3 TO 5
    PRINT "Ellipse #":Imat-3
    PRINT "Mean: (";Xbar(Imat);", ";Xbar(Imat);")
    PRINT "Covariance Matrix:
    PRINT ";Matrx(1,1,Imat);" ;Matrx(1,2,Imat);" **
    PRINT ";Matrx(2,1,Imat);" ;Matrx(2,2,Imat);" **
PRINT NEXT Imat
PRINT "Last Combined Ellipses"
PRINT ";Matrx(1,1,6);" ;Matrx(1,2,6);" **
PRINT ";Matrx(2,1,6);" ;Matrx(2,2,6);" **
PRINT "Simulation Results:
PRINT "># of simulations: ";Nsimulations
PRINT ". Totals; % of Total; % of Category"
PRINT ".Accepted: ";N_accept; ";100*N_accept/Nsimulations
PRINT ".Target 1 NOT in: ";Anotin1; ";100*Nnotin1/Nsimulations
PRINT ".Target 2 NOT in: ";Anotin2; ";100*Nnotin2/N_accept
PRINT 
Simulation Results:
PRINT "Neither one in: ";Anotin2/Nsimulations; "Neither one in: ";Anotinb/Nsimulations
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT "Neither one in: ";Anotinb/Nsimulations;
PRINT "Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " Neither one in: ";Anotin2/Nsimulations;
PRINT " Neither one in: ";Anotinb/Nsimulations;
PRINT " None of the differences in the Mean components:";
PRINT "Enter the differences in the Mean components:";
PRINT "Mean difference along the X axis: ";
INPUT "X Difference: ", Delta(1)
PRINT Delta(1)
PRINT "Mean difference along the Y axis: ";
INPUT "Y Difference: ", Delta(2)
PRINT Delta(2)
\% Calculate parameters\%
LET Lambda=0
FOR I=1 TO 2
  FOR J=1 TO 2
    Lambda=Lambda+Power_mat(I,J,2)*Delta(I)*Delta(J)
  NEXT J
NEXT I
PRINT "Lambda = "; Lambda
\% Linear Interpolation\%
GOSUB Linear_interp
PRINT "Linear Power: "; Li_power
\% Lagrange Interpolation\%
GOSUB Lagrange_interp
PRINT "Lagrange Power: "; La_power
\% Pause\%
IF Delta(1)<>0 OR Delta(2)<>0 THEN 8360
8610 RETURN
8620 LET Linear_interp: 1
8630 LET Linear_interp: 1
8640 LET Linear_interp: 1
\% Find Bounding Values\%
8650 LET Linear_interp: 1
8660 LET Linear_interp: 1
8670 \% Search\%
8680 IF Lambda>-39 THEN
8690 LET 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 LET 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 LET 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_l)/(Lambda_h-Lambda_l)
8890 LET Li_power=Power1+Power2
8900 \% Lagrange Interpolation\%
8910 \% Search\%
8920 \% Interpolation\%
END IF

Summer=0

FOR I=Search-3 TO Search+2
Prod=Table(I,2)
END IF

FOR J=Search-3 TO Search+2
IF J<>I THEN
Prod=Prod*(Lambda-Table(J,1))/(Table(I,1)-Table(J,1))
END IF

NEXT J

Summer=Summer+Prod

NEXT I

La_power=Summer

RETURN

Chi-square(2) Power Table. 1st column is non-centrality parameter, 2nd 3rd power. 0.05 significance label. From Selected Tables in
Mathematical Statistics, Volume 1.

Power data:

DATA .0,.05
DATA .1,.0576
DATA .2,.0653
DATA .3,.0733
DATA .4,.0814
DATA .5,.0896
DATA .6,.0980
DATA .7,.1065
DATA .8,.1151
DATA .9,.1239
DATA 1,.1327
DATA 1.2,.1507
DATA 1.4,.1691
DATA 1.6,.1877
DATA 1.8,.2065
DATA 2,.2255
DATA 2.2,.2447
DATA 2.4,.2639
DATA 2.6,.2831
DATA 2.8,.3024
DATA 3,.3215
DATA 3.5,.3690
DATA 4,.4154
DATA 4.5,.4604
DATA 5,.5037
DATA 6,.5480
DATA 7,.6154
DATA 8,.7176
DATA 9,.8707
DATA 10,.8154
DATA 11,.8526
DATA 12,.8832
DATA 13,.9080
DATA 14,.9280
DATA 15,.9400
DATA 16,.9567
DATA 17,.9667
DATA 18,.9745
DATA 19,.9805
DATA 20,.9852
DATA 21,.9888
DATA 22,.9916
DATA 23,.9937
DATA 24,.9953
DATA 25,.9965
DATA 26,.9974
DATA 27,.9981
DATA 28,.9986
DATA 29,.9989

B-14
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 39,1000
9800 DATA 40,1000
9810 DATA 41,1000
9820 DATA 42,1000
9830 End_sub2: 1
9840 SUBEND
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.
TEST PROGRAM FOR ELLIPSE ROUTINE LIBRARY

LIBRARY SUBROUTINES BEGIN ON LINE 1000.

THIS TEST PROGRAM GETS AN ELLIPSE FROM THE USER AND DISPLAYS IT ON
THE SCREEN.

GCLEAR

REAL Prob_constant

Prob_constant=2*LOG(.05)

Get_Ell_mean(1)

Get_covariance(1)

Get_axes(1)

Invert(1,21)

Get_bounds(1,Prob_constant)

Choose_plotter

Draw_ellipse(-50,50,-50,50,Prob_constant)

STOP

END

ELLIPSE ROUTINE LIBRARY....

This file contains standard ellipse routines and data variables,
including the following:

- Entry of ellipses by covariance matrices.
- Entry of ellipses by axes and orientation.
- Display of ellipses on screen and plotter.
- matrix inversion routine for use with Matrx.
- Matrx, an array which stores covariance matrices and their
  inverses (up to 10 matrices).
- Axes/orientation to Covariance matrix conversion routine.

USING MATRX

Matrx is designed to hold covariance matrices and their inverses
for 10 ellipses. In general, Matrx(1)...Matrx(20) are the covariance
matrices and Matrx(21)...Matrx(40) are the corresponding inverses.
This is the convention assumed by a number of these routines.
In cases where the inverse is calculated first, put in Matrx(25), say,
and then call Invert(25, 5). This will put the covariance matrix in
Matrx(5).

THE ROUTINES ARE CALLED AS FOLLOWS:

Invert(Srce,Dest)

This command will invert the covariance matrix in Matrx(Srce) and
put the result in Matrx(Dest)

Get_Bounds(Ellipse,Prob_constant)

This command will get the extreme points of the ellipse. Note that
the ellipse's covariance matrix must have been inverted.

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

This will draw the given ellipse. The remaining variables define
the screen. Note that Get_bounds must have been executed.

Get_Ell_Mean(Ellipse)

This is an input routine to read in Xbar and Ybar for the given
ellipse.

Get_Covariance(Ellipse)

This is an input routine to read in the covariance matrix for the
given ellipse.

Get_Axes(Ellipse)

This is an input routine which reads in the axes and orientation
of the ellipse, and converts this to covariance matrix form.

Choose_Plotter

This prompts the user to draw the ellipse on the plotter or the
screen.

Display_Cov(Ellipse)

This command displays the given covariance matrix.

Test(Ellipse1,Ellipse2,Work,Test_Stat)

This calculates the acceptance test criteria for the two ellipses.

Work is a Matrx entry used as working space.

Add_Covariance(Ellipse1,Ellipse2,Summer)
This adds any two matrix entries into a third matrix entry.

Combine Ellipse(Ellipse1, Ellipse2, Combo)

This combines any two matrix entries into a third matrix entry.

Display Extremes(Ellipse)

This displays the extreme X and Y bounds for the ellipse.

Also included are Pausebit and Clearscreen.

This routine inverts any covariance matrix in Matrix and places
the inverted matrix in Dest.

COM /Ellipses/ Matrix(2,2,40), Xbar(20), Ybar(20), Major_axis(20), Minor_axis(20), Theta(20), Angle(20), Emin_x(20), Emin_y(20), Emx_x(20), Emx_y(20)

Det=Matrix(1,1,Src)-Matrix(2,2,Src)*Matrix(1,2,Src)+Matrix(1,2,Src)*Matrix(1,1,Src)/Det

Matrix(1,1,Dest)=Matrix(2,2,Src)/Det

Matrix(1,2,Dest)=Matrix(1,2,Src)/Det

Matrix(2,1,Dest)=Matrix(2,1,Src)/Det

SUBEND ! End of SUB Invert

SUB Get_bounds(Ellipse, Prob constant)

This subroutine calculates the X and Y limits for the given ellipse

COM /Ellipses/ Matrix(2,2,40), Xbar(20), Ybar(20), Major_axis(20), Minor_axis(20), Theta(20), Angle(20), Emin_x(20), Emin_y(20), Emx_x(20), Emx_y(20)

REAL Temp1, Temp2, Temp3, Temp4, Temp5

Temp1=Matrix(2,2,Ellipse)*Prob_constant

Temp2=Matrix(1,1,Ellipse)*Prob_constant

Temp3=(Temp1/Temp2)^.5

Temp4=Matrix(1,1,Ellipse)*Prob_constant

Temp5=(Temp4/Temp2)^.5

Emin_y(Ellipse)=Temp5*Ybar(Ellipse)

Emx_y(Ellipse)=Temp5*Xbar(Ellipse)

Emin_x(Ellipse)=Temp3*Ybar(Ellipse)

Emx_x(Ellipse)=Temp3*Xbar(Ellipse)

SUBEND ! End of GET_BOUNDS

SUB Choose_plotter

SUBroutine to choose the desired plotter device (also contained

in GRAPHER).

REPEAT

Go_on=1

DISP "Do you want to (D)raw, (E)xclude, Plot with Pen (0), (1), (2)?"

LINPUT Message$

SELECT Message$

CASE "D", "d"

PLOTTER IS 3, "INTERNAL"

GRAPHICS ON

PEN 1

Go_on=1

CASE "0"

PLOTTER IS 705, "HPGL"

GRAPHICS ON

PEN 0

Go_on=1

CASE "1"

PLOTTER IS 3, "INTERNAL"

GRAPHICS ON

PEN 1

Go_on=1

CASE "2"

PLOTTER IS 705, "HPGL"

GRAPHICS ON
1944  PEN 2
1945  Go on=
1940  CASE Type="E"
1960  PLOTTER IS 3,"INTERNAL"
1970  GRAPHICS ON
1980  PEN 1
1990  Go on=
2000  CASE ELSE
2010  PRINT CHR$(7)
2020  END SELECT
2030  UNTIL Go on=
2040  SUBEND
2050  SUB Get_all_mean(Ellipse)
2100  CON /Ellipses/ Matrix(2,2,40), Xbar(20), Ybar(20), Major_axis(20), Minor_axis(20), Theta(20), Angle(20), Emax_x(20), Emin_y(20), Emx_y(20)
2110  Clearscreen
2120  PRINT TABXY(1,10);"WHAT IS THE MEAN FOR ELLIPSE "; Ellipse; ";?"
2130  INPUT Xbar(Ellipse)
2140  PRINT Xbar(Ellipse)
2150  PRINT TABXY(1,11);"WHAT IS Y-BAR FOR ELLIPSE "; Ellipse; ";?"
2160  INPUT Ybar(Ellipse)
2170  PRINT Ybar(Ellipse)
2180  Pauseabit
2190  SUBEND
2230  SUB Get_covariance(Ellipse)
2240  ; This subroutine gets the covariance matrix for an ellipse
2250  CON /Ellipses/ Matrix(2,2,40), Xbar(20), Ybar(20), Major_axis(20), Minor_axis(20), Theta(20), Angle(20), Emax_x(20), Emin_y(20), Emx_y(20)
2260  Clearscreen
2270  FOR K=1 TO 2
2280  PRINT TABXY(1,3+K-6);"ENTER ELEMENT ("; K; "; "; K; "); IN THE COVARIANCE MATRIX FOR ELLIPSE "; Ellipse; ";:
2290  INPUT Matrix(K,K,Ellipse)
2300  PRINT Matrix(K,K,Ellipse)
2310  NEXT K
2320  PRINT TABXY(1,3+9);"ENTER ELEMENT (1,2) IN THE COVARIANCE MATRIX FOR ELLIPSE "; Ellipse; ";:
2330  INPUT Matrix(1,2,Ellipse)
2340  Matrix(2,1,Ellipse)=Matrix(1,2,Ellipse)
2350  PRINT Matrix(1,2,Ellipse)
2360  Pauseabit
2370  SUBEND
2410  SUB Pauseabit
2420  ; Pause and wait for a carriage return
2430  DISP "Type ENTER to continue.";
2440  INPUT Garbage$<CR>
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  CON /Ellipses/ Matrix(2,2,40), Xbar(20), Ybar(20), Major_axis(20), Minor_axis(20), Theta(20), Angle(20), Emax_x(20), Emin_y(20), Emx_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; "; : ";
INPUT Minor_axis(Ellipse)
PRINT Minor_axis(Ellipse)
PRINT "WHAT IS THE ANGLE (IN DEGREES) OF THE MAJOR AXIS OF ELLIPSE? ;E Ellipse": ";
INPUT Angle(Ellipse)
PRINT Angle(Ellipse)
CONVERT TO COVARIANCE MATRIX
Theta(Ellipse)=PI*Angle(Ellipse)/180
Matrx(1,1,Ellipse)=((Major_axis(Ellipse))*COS(Theta(Ellipse)))*2-(Minor_axis(Ellipse))*SIN(Theta(Ellipse)))*2
Matrx(2,2,Ellipse)=((Major_axis(Ellipse))*COS(Theta(Ellipse)))*2-(Minor_axis(Ellipse))*SIN(Theta(Ellipse)))*2
Matrx(2,1,Ellipse)=((Major_axis(Ellipse)))*2-(Minor_axis(Ellipse))*COS(Theta(Ellipse)))*2
Matrx(1,2,Ellipse)=Matrx(2,1,Ellipse)
FOR J=1 TO 2
FOR K=1 TO 2
Matrx(J,K,Ellipse)=Matrx(J,K,Ellipse)/(-Prob_Constant)
NEXT K
NEXT J
PRINT
Pauseabit
SUBEND
SUB Display_cov(Ellipse)
COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Em_in(20),Em_max(20),Em_in(20),Em_max(20)
FOR I=1 TO 2
FOR K=1 TO 2
PRINT Matrx(I,J,Ellipse); ";
NEXT J
NEXT I
PRINT
SUBEND
SUB Add_covariance(First,Second,Summer)
COM /Ellipses/ Matrx(2,2,40),Xbar(20),Ybar(20),Major_axis(20),Minor_axis(20),Theta(20),Angle(20),Em_in(20),Em_max(20),Em_in(20),Em_max(20)
INTEGER I,J
FOR I=1 TO 2
FOR J=1 TO 2
Matrx(I,J,Summer)=Matrx(I,J,First)+Matrx(I,J,Second)
NEXT J
NEXT I
SUBEND
SUB Combine_ellipse(First,Second,Combo)
REAL Sx1,Sx2
I Calculate the new covariance matrix.
Invert(First,First+20)
Invert(Second,Second+20)
Add_covariance(First+20,Second+20,Combo+20)
I Calculate the new mean point.
Sx1=Matrx(1,1,First+20)*Xbar(First)+Matrx(1,2,First+20)*Ybar(First)
Sx2=Matrx(2,1,First+20)*Xbar(First)+Matrx(2,2,First+20)*Ybar(First)
Sx2=Sx2+Matrx(2,1,Second+20)*Xbar(Second)+Matrx(2,2,Second+20)*Ybar(Second)
Sx1=Sx1+Matrx(1,1,Combo+20)*Xbar(Combo)+Matrx(1,2,Combo+20)*Ybar(Se cond)
Sx2=Sx2+Matrx(2,1,Combo+20)*Xbar(Combo)+Matrx(2,2,Combo+20)*Ybar(Se cond)
Xbar(Combo)=Matrx(1,1,Combo)*Sx1+Matrx(1,2,Combo)*Sx2
Ybar(Combo)=Matrx(2,1,Combo)*Sx1+Matrx(2,2,Combo)*Sx2
SUBEND
This routine displays the extreme x and y values for the specified ellipse.

CON /Ellipses/ Matrix(2,2,40), Xbar(20), Ybar(20), Major_axis(20), Minor_axis(20), Theta(20), Angle(20), Emin_x(20), Emx_x(20), Emin_y(20), Emx_y(20)

PRINT "Ef" ;Ellipse;
PRINT "Xmin" ;Emx_x(Ellipse);
PRINT "Xmax" ;Emx_x(Ellipse);
PRINT "Ymin" ;Emx_y(Ellipse);
PRINT "Ymax" ;Emx_y(Ellipse);
SUBLEND
SUBLEND=0
SUBLEND=5

This routine draws the ellipse on the current plotter device.

INTEGER Sign
REAL Xpoint, Upoint, Temp1, Temp2, Temp3, Temp5
Get_bounds(Ellipse)
Invert(Ellipse, Ellipse+20)

GRAPHICS ON

DRAW (Xbar(Ellipse)-Xmin)*100/(Xmax-Xmin), (Ybar(Ellipse)-Ymin)*100/(Ymax-Ymin)

RETURN
SUBEND

This routine calculates the acceptance test criteria for First and Second. Work is used as a "scratchpad".

CON /Ellipses/ Matrix(2,2,40), Xbar(20), Ybar(20), Major_axis(20), Minor_axis(20), Theta(20), Angle(20), Emin_x(20), Emx_x(20), Emin_y(20), Emx_y(20)
Add_covariance(First, Second, Work)
Invert(Work, Work+20)
Diffs = Xbar(First) - Xbar(Second)
Diffs = Ybar(First) - Ybar(Second)
Temp1 = Matrix(1,1, Work+20) * Diffs * Matrix(1,2, Work+20) * Diffs
Temp2 = Matrix(2,1, Work+20) * Diffs * Matrix(2,2, Work+20) * Diffs
Test_val = Diffs * Temp1 + Diffs * Temp2
SUBEND
END

DTIC

5-86