NASA Technical Memorandum 100730

User's Guide Programs for Processing Altimeter Data Over Inland Seas

A. Y. Au, R. D. Brown
and J. E. Welker

April 1989

NASA

NASA Technical Memorandum 100730

User's Guide Programs for Processing Altimeter Data Over Inland Seas

A. Y. Au and R. D. Brown
ST Systems Corporation (STX)
Lanham, Maryland

J. E. Welker
Geodynamics Branch
Laboratory for Terrestrial Physics
Goddard Space Flight Center
Greenbelt, Maryland

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

1989

PREFACE

Satellite-based altimetric data taken by GEOS-3 and SEASAT over the Black Sea and Caspian Sea have been analyzed and a least-squares collocation technique used to predict the geoid undulations on a .25° x .25° grid and to transform these geoid undulations to free air gravity anomalies. The programs used to process and analyze these altimeter data are described and documented herein. A brief user's guide is presented for each program that summarizes the purpose for, input to, and output from the code. This is followed by a complete listing of well commented code in the FORTRAN-77 language. These programs read, organize, and plot profiles and ground tracks of the altimeter data. They also identify crossovers between altimeter passes, determine and remove biases from the data, sort, and grid the corrected altimetric sea surface heights. Other programs generate contour plots of the gridded data, determine the gravity and geoid auto- and cross-covariance functions, and use these functions in a collocation estimate of gravity anomalies.

TABLE OF CONTENTS

I INTRODUCTION

The programs described in this document have been developed to process GEODYN-formatted satellite altimeter data, and to apply the processed results to predict geoid undulations and gravity anomalies of inland sea areas (Au *et al.*, 1988). These programs are written in standard FORTRAN 77 and are designed to run on the NSESCC IBM 3081 (MVS) computer. Because of the experimental nature of these programs, they are tailored to the geographical area analyzed. The attached program listings are customized for processing the altimeter data over the Black Sea. Users interested in the Caspian Sea data are expected to modify each program, although the required modifications are generally minor. Program control parameters are defined in the programs via PARAMETER statements and/or DATA statements. Other auxiliary parameters, such as labels, are 'hard-wired' into the programs. Large data files are read in or written out through different input or output units. The program listings of these programs are accompanied by sample IBM Job Control Language (JCL) images. Familiarity with IBM JCL and the TEMPLATE graphic package is assumed.

II USER'S GUIDES

PRTDATA

This program reads a GEODYN-formatted altimeter data file, groups together the data records that belong to a satellite pass and extracts relevant geodetic and auxiliary information from each satellite pass. The process is repeated for all the passes in the input data file. For each satellite pass, the starting time in Julian day, the satellite identification number, the duration in seconds, the direction of the data arc (ascending or descending in latitude) and the number of data records associated with the pass are printed. For each data record, the corresponding record number in the file, the fraction of day in which the measurement is taken, the absolute altimeter observation (meters), the ground track geodetic latitude and longitude and the observed surface elevation (meters) are printed. If a reference geoid undulation surface is supplied to the program, a comparison between the reference geoid undulation and the measured one is given for each data record.

User-provided Parameters:

MAXREC	Maximum number of records in the input altimeter data file.
XMIN	Starting longitude of the input reference geoid undulation surface.
XMAX	Ending longitude of the input reference geoid undulation surface.
YMIN	Starting latitude of the input reference geoid undulation surface.
YMAX	Ending latitude of the input reference geoid undulation surface.
DX	Increment in longitude of the input geoid reference undulation surface.
DY	Increment in latitude of the input geoid reference undulation surface.

Input:

Reference geoid undulation surface (Unit 5):

The input file should be organized as follows:

starting geodetic latitude YMIN (unformatted).
geoid undulations of associated longitude from XMIN to XMAX
incremented by DX (6(2X,F10.3)).
.
.
YMIN + DY.
geoid undulations of associated longitude from XMIN to XMAX incremented by DX.
.
.
ending geodetic latitude YMAX.
geoid undulations of associated longitude from XMIN to XMAX incremented by DX.

Altimeter data file (Unit 8): This is a binary file of altimeter data written in GEODYN format.

Output:

Organized printout (Unit 6).

Reference geoid undulation corresponding to each data record (Unit 9): Each output record contains latitude, longitude and reference geoid undulation (5X,2F9.3,45X,F9.3). This output file may be used in subsequent programs for the purpose of comparison between measured geoid undulations and reference values.

PLTGRP

This program reads a GEODYN-formatted altimeter data file, groups together the data records that belong to a satellite pass and plots the surface elevation profile of each pass as a function of time (fraction of day past midnight). The process is repeated for all the passes in the input data file. A surface elevation profile will not be plotted if there are less than four records in the satellite pass. The graphic software package, TEMPLATE, is used with this program.

Input:

Altimeter data file (Unit 5): This is a binary file of altimeter data written in GEODYN format.

Output:

Summary (Unit 6): Number of records read in input data file and the number of surface elevation profiles plotted.

Plots of surface elevation profiles of each satellite pass as a function of time.

GRNTRK

This program reads a GEODYN-formatted altimeter data file, groups together the data records that belong to a satellite pass and plots the ground track onto a geographical map. The process is repeated for all the passes in the input data file. A graphic software package, TEMPLATE, is used in this program.

User-provided Parameters:

MAXREC	Maximum number of records in the input altimeter data file.
NMAX	Maximum number of data records associated with one satellite pass.
VLONG1	Starting east longitude of the output geographical map.
VLONG2	Ending east longitude of the output geographical map.
VLAT1	Starting latitude of the output geographical map.
VLAT2	Ending latitude of the output geographical map.

Input:

Altimeter data file (Unit 5): This is a binary file of altimeter data written in GEODYN format.

Output:

Summary (Unit 6): Number of records read in the input file, number of GEOS-3 and SEASAT ground tracks plotted.

Ground tracks plot.

XOVER0

This program reads a GEODYN-formatted altimeter data file, approximates the ground track of each satellite pass as a second-degree equation and determines potential crossovers of these satellite passes. An overview of satellite crossovers is provided by this program before crossover adjustments are performed in a subsequent program, XOVER. Crossovers that occur at data gaps can be edited out.

User-provided Parameters:

MPASS	Maximum number of satellite passes in the input altimeter data file.
XMAX	Maximum number of data records associated with a satellite pass.
SIGE	Discriminating criterion of a data spike. Observed surface elevation that deviates from adjacent one by more than SIGE will be treated as noise and, thereby, will not be included in subsequent data analyses.

Input:

Altimeter data file (Unit 3): This is a binary file of altimeter data written in GEODYN format.

Output:

Summary (Unit 6): Possible satellite crossovers, including satellite types, crossover locations.

XOVER

Crossover adjustment is performed in this program. This program reads a GEODYN-formatted altimeter data file, approximates each satellite pass ground track by a second-degree equation and determines possible crossovers of these satellite passes. The geoid height at each crossover location for each pass is obtained by interpolating screened data in that pass. A weighted least-squares analysis is performed to determine biases for each pass so that the crossover residuals are minimized, holding one pass fixed as the reference pass. All satellite passes are then rectified to a common reference geoid through the chosen reference pass. The output includes the RMS of crossover residual before and after orbit bias adjustment, a GEODYN-formatted file with the original observed surface elevation replaced by the adjusted one, and an error covariance matrix of the bias adjustment.

User-provided Parameters:

SIGMAG	Assumed noise for GEOS-3 orbit.
SIGMAS	Assumed noise for SEASAT orbit.
XDMAX	Maximum crossover residual for that crossover to be considered.
ETOP	Upper limit of observed surface elevation to be accepted.
EBOT	Lower limit of observed surface elevation to be considered.
MAXOVR	Maximum number of crossovers.
MPASS	Maximum number of satellite passes in the input altimeter data file.
NMAX	Maximum number of data records associated with a satellite pass.
SIGE	Discrimination criterion of a data spike. Observed surface elevation that deviates from adjacent one by more than SIGE will be treated as noise and, thereby, will not be included in subsequent data analyses.

Input:

Reference geoid undulation model (Unit 2): Each record in the file contains latitude,

longitude and reference geoid undulation (5X,2F9.3,45X,F9.3). This is the output file from program PRTDATA.

Altimeter data file (Unit 3): This is a binary file of altimeter data written in GEODYN format.

Crossovers to be edited out (Unit 4): This file contains records of two passes that are to be excluded from crossover adjustment. Each record contains three variables, JC, IP, and JP, where IP and JP are the pass number of the two passes that form the crossover, and JC is an editing criterion. If JC is set to zero, that crossover will be excluded from crossover adjustment. The format of the record is (I1,2I10).

Output:

Adjusted altimeter data file (Unit 8): This is a binary file written in GEODYN format with the observed surface elevation replaced by the adjusted one.

Error covariance matrix of the bias adjustment (Unit 9): The format of this output matrix is (4D20.13).

PLTAEP

This program reads an adjusted altimeter data file in GEODYN format and a file containing reference geoid undulation that is in one-to-one correspondence with the adjusted altimeter data, and plots the adjusted elevation profile of each satellite pass. The reference geoid is also plotted onto each elevation profile for comparison. A graphic software package, TEMPLATE, is used in this program.

User-provided Parameters:

MAXPLT	Maximum number of plots to be performed.
ETOP	Upper limit of observed surface elevation to be accepted.
EBOT	Lower limit of observed surface elevation to be considered.
NMAX	Maximum number of data records associated with a satellite pass.
NINT	Number of intervals used in a smoothing routine to plot out a smoothed elevation profile of a satellite pass.
SIGE	Discriminating criterion of a data spike. Observed surface elevation that deviates from adjacent one by more than SIGE will be treated as noise and, thereby, will not be included in subsequent data analyses.
SIGT	Not used.

Input:

Reference geoid undulation model (Unit 3): Each record in the file contains latitude, longitude and reference geoid undulation (5X,2F9.3,45X,F9.3). This is the output file from program PRTDATA.

Altimeter data file (Unit 4): This is a binary file of altimeter data written in GEODYN format.

Output:

Plots of altimeter surface elevation profiles adjusted for orbit biases. A reference geoid is superimposed on each profile for comparison.

SORT

This program reads an adjusted altimeter data file and determines with which grid points a data record should be associated, according to a specified cap size. Each GEODYN data record is checked for grid point association. For each grid point associated with a record, an output record is written with latitude, longitude, adjusted surface elevation, grid points identification number, grid point counter, associated pass number, and standard deviation of that data point. A sort/merge process (TSORT in MVS system) can be performed to group together all the altimeter data points that are associated to each grid point.

User-provided Parameters:

ETOP	Upper limit of adjusted surface elevation to be accepted.
EBOT	Lower limit of adjusted surface elevation to be considered.
SIGE	Discriminating criterion of a data spike. Observed surface elevation that deviates from adjacent one by more than SIGE will be treated as noise and, thereby, will not be included in subsequent data analyses.
MIREC	Maximum number of data records associated with a satellite pass.
XMIN	Starting longitude of the chosen grid.
XMAX	Ending longitude of the chosen grid.
YMIN	Starting latitude of the chosen grid.
YMAX	Ending latitude of the chosen grid.
DX	Increment in longitude of the chosen grid.
DY	Increment in latitude of the chosen grid.
CAP	Cap size within which altimeter data is considered.

Input:

Altimeter data file (Unit 5): This is a binary file of adjusted altimeter data written in GEODYN format.

Output:

Sunimary (Unit 6): Data partitioning information and data distribution map.

Altimeter data partition file (Unit 8): Each record in this binary file contains the longitude, latitude, adjusted surface elevation, grid identification index for longitude, grid identification index for latitude, satellite pass number, and standard deviation of the adjusted surface elevation. In order to be used in subsequent Program GEOID, this output file should be sorted in ascending order of the grid longitude index and grid latitude index.

WGTAVG

This program reads an adjusted altimeter data file written in GEODYN format, sorts each data record into a .25° x .25° grid, and determines a weighted average of the data points at each grid point.

User-provided Parameters:

ETOP	Upper limit of adjusted surface elevation to be accepted.
EBOT	Lower limit of adjusted surface elevation to be considered.
SIGE	Discrimination criterion of a data spike. Observed surface elevation that deviates from adjacent one by more than SIGE will be treated as noise and, thereby, will not be included in subsequent data analyses.
MIREC	Maximum number of data records associated with a satellite pass.
XMIN	Starting longitude of the chosen grid.
XMAX	Ending longitude of the chosen grid.
YMIN	Starting latitude of the chosen grid.
YMAX	Ending latitude of the chosen grid.
DX	Increment in longitude of the chosen grid.
DY	Increment in latitude of the chosen grid.
MXP	Maximum number of data points associated with a grid point.

Input:

Altimeter data file (Unit 5): This is a binary file of adjusted altimeter data written in GEODYN format.

Output:

Summary (Unit 6): Data partitioning information, weighted averages and data distribution map.

Gridded weighted-averages (Unit 8): The output file contains sequences of grid longitude, grid latitude and surface elevation. The format of the output file is (3(2F7.2,F12.4).

CONTOUR

This program reads a gridded data file and plots the corresponding contour. A geographical map defined within the chosen grid boundary is superimposed on the contour. Any grid point that has no physical value should be masked with the value of -1000.0 and it will not be contoured. A graphic software package, TEMPLATE, is used in this program.

User-provided Parameters:

BIAS	Overall bias that is to be subtracted off all the data points.
XMIN	Starting longitude of the chosen grid.
XMAX	Ending longitude of the chosen grid.
YMIN	Starting latitude of the chosen grid.
YMAX	Ending latitude of the chosen grid.
DX	Increment in longitude of the chosen grid.
DY	Increment in latitude of the chosen grid.
CHGH	Upper limit of contour.
CLOW	Lower limit of contour.
CINC	Increment interval of contour.
CLAB	Frequency of labelling the contour lines.

Input:

Gridded data file (Unit 5): The output file contains sequences of grid longitude, grid latitude and surface elevation. The format of the output file is (3(2F7.2,F12.4).

Output:

A contour map of the gridded data.

GEOID

This program predicts and grids geoid undulations using a collocation technique. The input dataset is a sorted version, according to grid longitude and latitude indices, of the output file from Program SORT.

User-provided Parameters:

MPASS	Number of satellite passes considered in the adjusted altimeter dataset.
NREFP	Relative position number of the reference satellite pass in the altimeter dataset.
KUTOFF	Minimum number of data points associated with a grid point to be considered for geoid prediction.
LCTV	Set to 1.
MTR	Maximum number of data records associated with a grid point.
XMIN	Starting longitude of the chosen grid.
XMAX	Ending longitude of the chosen grid.
YMIN	Starting latitude of the chosen grid.
YMAX	Ending latitude of the chosen grid.
DX	Increment in longitude of the chosen grid.
DY	Increment in latitude of the chosen grid.
LCT	Number of intervals in the given covariance function.

Input:

Covariance table (Unit 5): Each record in the covariance table contains distance in degree, geoid-to-geoid value, geoid-to-gravity value, and gravity-to-gravity value. The format of the input file is (1X,D10.4,3D20.13).

Sorted altimeter data file (Unit 8): This is a sorted version of the binary output file from Program SORT. The sorting should be performed in ascending order of grid longitude index and then of grid latitude index.

Error covariance matrix of orbit adjustment (Unit 11): This is the output error covariance matrix from Program XOVER.

Output:

Gridded geoid undulations (Unit 9): The output file contains sequences of grid longitude, grid latitude and surface elevation. The format of the output file is (3(2F7.2,F12.4).

Gridded variance of predictions (Unit 10): The output file contains sequences of grid longitude, grid latitude and variance in cm. The format of the output file is (3(2F7.2,F12.4).

EMPCOV

This program determines a set of residual local empirical covariance functions based on a convolution technique. Reference models as well as gridded data of both geoid undulations and gravity anomalies must be supplied to the program.

User-provided Parameters:

XMIN	Starting longitude of the chosen grid.
XMAX	Ending longitude of the chosen grid.
YMIN	Starting latitude of the chosen grid.
YMAX	Ending latitude of the chosen grid.
DXY	Increment in longitude and latitude of the chosen grid.

Input:

Reference geoid undulation surface (Unit 7):

The input file should be organized as follows:

starting geodetic latitude YMIN (unformatted).
geoid undulations of associated longitude from XMIN to XMAX incremented by DXY (6(2X,F10.3)).
.
.
YMIN + DXY.
geoid undulations of associated longitude from XMIN to XMAX incremented by DX.
.
ending geodetic latitude YMAX.
geoid undulations of associated longitude from XMIN to XMAX incremented by DX.

Reference gravity anomalies surface (Unit 9):

Input format similar to Unit 7.

Gridded geoid undulations (Unit 8): The input file contains sequences of grid longitude, grid latitude and surface elevation. The format of the input file is (3(2F7.2,F12.4).

Gridded gravity anomalies (Unit 10): The input file contains sequences of grid

longitude, grid latitude and gravity anomalies. The format of the input file is (3(2F7.2,F12.4).

Output:

Covariance table (Unit 11): Each record in the covariance table contains distance in degree, geoid-to-geoid value, geoid-to-gravity value, and gravity-to-gravity value. The format of the output file is (1X,D10.4,3D20.13).

GRAVEN

This program transforms gridded geoid undulation data into gravity anomalies. Three reference models, Rapp's 36 x 36, 180 x 180, and 300 x 300 gravity models (Rapp, 1986) can be used as reference surfaces in this program. An iterative process to refine the reference surfaces can also be performed by this program.

User-provided Parameters:

MODEL	Reference model to be used. It can be 36, 180 or 300, which corresponds to Rapp's 36 x 36, 180 x 180, and 300 x 300 models.
ANOISE	Assumed noise of the geoid undulation data.
ITM	Maximum number of iterations to be carried out. When there is no iteration to be performed, ITM is set to zero.
CAP	Integration cap size used.
IPRT	Option for printouts. Set IPRT to zero for minimum output.
CNVRGE	Convergence criterion for iterative process.
SCX	Set to 1.
KUTOFF	Minimum number of data points associated with a grid point to be considered for gravity transformation.
MTR	Maximum number of data records associated with a grid point.
XMIN	Starting longitude of the chosen grid.
XMAX	Ending longitude of the chosen grid.
YMIN	Starting latitude of the chosen grid.
YMAX	Ending latitude of the chosen grid.
XG1	Set equal to XMIN.
XG2	Set equal to XMAX.
YG1	Set equal to YMIN.
YG2	Set equal to YMAX.

DX Increment in longitude of the chosen grid.

DY Increment in latitude of the chosen grid.

Input:

Covariance table (Unit 5): Each record in the covariance table contains distance in degree, geoid-to-geoid value, geoid-to-gravity value, and gravity-to-gravity value. The format of the input file is (1X,D10.4,3D20.13).

Gridded geoid undulations (Unit 8): The input file contains sequences of grid longitude, grid latitude and surface elevation. The format of the input file is (3(2F7.2,F12.4).

Gridded variance of geoid prediction (Unit 9): The input file contains sequences of grid longitude, grid latitude and variance in cm. The format of the input file is (3(2F7.2,F12.4).

Reference geoid undulation surface, Rapp's 180 x 180 model (Unit 10):

The input file should be organized as follows:

starting geodetic latitude YMIN (unformatted).
geoid undulations of associated longitude from XMIN to XMAX incremented by DX (6(2X,F10.3)).

.

.

YMIN + DY.
geoid undulations of associated longitude from XMIN to XMAX incremented by DX.

.

ending geodetic latitude YMAX.
geoid undulations of associated longitude from XMIN to XMAX incremented by DX.

Reference gravity anomalies surface, Rapp's 180 x 180 model (Unit 11):

Input format similar to Unit 10.

Reference geoid undulation surface, Rapp's 36 x 36 model (Unit 12):

Input format similar to Unit 10.

Reference gravity anomalies surface, Rapp's 36 x 36 model (Unit 13):

Input format similar to Unit 10.

Reference geoid undulation surface, Rapp's 300 x 300 model (Unit 14):

Input format similar to Unit 10.

Reference gravity anomalies surface, Rapp's 300 x 300 model (Unit 15):

Input format similar to Unit 10.

Mask of inland sea surface (Unit 16): This is an M x N mask of the inland sea surface, where M and N are the numbers of longitude and latitude grid points. A unitary value (1) indicates that a prediction and transformation are to be performed at that grid point. A zero (0) indicates that no prediction or transformation is to take place at that grid point.

Output:

Gridded gravity anomalies (Unit 17): The output file contains sequences of grid longitude, grid latitude and gravity anomalies. The format of the output file is (3(2F7.2,F12.4).

Gridded variance of gravity transformation (Unit 18): The output file contains sequences of grid longitude, grid latitude and variance in mgal. The format of the output file is (3(2F7.2,F12.4).

Gridded geoid undulations (Unit 19): The output file contains sequences of grid longitude, grid latitude and geoid undulations. The format of the output file is (3(2F7.2,F12.4).

Gridded variance of geoid prediction (Unit 20): The output file contains sequences of grid longitude, grid latitude and variance in cm. The format of the output file is (3(2F7.2,F12.4).

Gridded difference between gravity anomalies at the end of the iterative process and the reference surface used (Unit 21): The output file contains sequences of grid longitude, grid latitude and gravity anomalies. The format of the output file is (3(2F7.2,F12.4).

Covariance table (Unit 22): This is a set of residual local empirical covariance functions obtained at the end of the iterative process. Each record in the covariance table contains distance in degree, geoid-to-geoid value, geoid-to-gravity value, and gravity-to-gravity value. The format of the output file is (1X,D10.4,3D20.13).

JORDAN

This program determines a set of theoretical covariance functions according to Jordan's formation (Jordan, 1972).

User-provided Parameters:

SN	Variance of geoid undulations at prediction point.
SG	Variance of gravity anomalies at prediction point.
D	Correlation distance.
DR	Increment in distance in degrees.
RMIN	Starting value of the distance in degrees.
RMAX	Ending value of the distance in degrees.

Output:

Covariance table (Unit 8): Each record in the covariance table contains distance in degree, geoid-to-geoid value, geoid-to-gravity value, and gravity-to-gravity value. The format of the output file is (1X,D10.4,3D20.13).

III. REFERENCES

Au, A. Y., R. D. Brown and J. E. Welker (1988) *Analysis of Altimetry Over Inland Seas*. **NASA Technical Memorandum - 100729.**

Jordan, S. K. (1972) *Self-constant statistical models for gravity anomaly, vertical deflections, and undulation of the geoid.* **J. Geophys. Res.**, 77, 3660-3670.

Rapp, R. H. (1986) *Gravity anomalies and sea surface heights derived from a combined GEO-3/SEASAT altimeter data set.* **J. Geophys. Res.**, 91, 4867-4876.

IV LISTINGS

A listing of PROGRAM PRTDATA

```
//ZMAYABSR JOB (G0109,360,2),AYAU,TIME=(0,30),CLASS=O,MSGCLASS=X
/*JOBPARM LINES=60
// EXEC FORTVC
//SYSIN DD *
C
C FORMAT OF GEOS-3/SEASAT ALTIMETER DATA
C
C VARIABLE     TYPE     DESCRIPTION
C
C      *         I4      SATELITE ID
C                I2      MEASUREMENT TYPE ( 42= OVER LAND , 43 = OVER WATER
C                I2      TIME SYSTEM ( NM )
C                I4      STATION NUMBER
C                I4      PREPROCESSING INDICATORS
C      *         I4      MODIFIED JULIAN DATE OF OBSERVATION
C      *         R8      FRACTION OF DAY PAST MIDNIGHT     (GMT)
C      *         R8      ALTIMETER OBSERVATION             (METERS)
C      *         I4      SATELLITE GEODETIC LATITUDE       (1E-6 DEGREES)
C      *         I4      SATELLITE EAST LONGITUDE          (1E-6 DEGREES)
C      *         R4      MEASUREMENT STANDARD DEVIATION    (METERS)
C                I4      NET INSTRUMENT CORRECTION         (MM)
C                I4      METEOROLOGICAL DATA WORD          (GEODYN VOL 3)
C                I4      NET MEDIA CORRECTIONS             (MM)
C                R4      GEOID HEIGHT ABOVE REFERENCE ELLIPSOID (METERS)
C                I4      NET OCEAN DYNAMIC CORRECTIONS     (MM)
C      *         I4      INDICATED SURFACE ELEVATION       (MM)
C      *         I4      S/C REVOLUTION NUMBER
C                I4      MEAN SEA SURFACE ELEVATION (MARSH/MARTIN '81 (MM))
C                I4      DOD REFERENCE RADIAL ORBIT DIFFERENCE (MM)
C                I2      H 1/3                             (CM)
C                I2      AGC                               (DB)
C                I2      WIND SPEED                        (CM/SEC)
C                I2      SURFACE ELEVATION PREPROCESSING WORD
C                I2      DRY TROPOSPHERIC CORRECTION       (MM)
C                I2      FNOC WET TROPOSPHERIC CORRECTION  (MM)
C                I2      SMMR WET TROPOSPHERIC CORRECTION  (MM)
C                I2      IONOSPHERIC CORRECTION            (MM)
C                I2      BAROTROPIC DYNAMIC SEA SURFACE CORRECTION (MM)
C                I2      SOLID EARTH TIDE                  (MM)
C                I2      SCHWIDERSKI OCEAN TIDE            (MM)
C                I2      PARKE OCEAN TIDE                  (MM)
C
C

      PARAMETER ( MAXREC = 10000 )
      PARAMETER ( XMIN = 26.5 )
      PARAMETER ( XMAX = 42.5 )
      PARAMETER ( YMIN = 40.0 )
      PARAMETER ( YMAX = 48.0 )
      PARAMETER ( DX = 0.25 )
      PARAMETER ( DY = 0.25 )
      PARAMETER ( NLON = ( (XMAX-XMIN)/DX + 1 ) )
      PARAMETER ( NLAT = ( (YMAX-YMIN)/DY + 1 ) )
      PARAMETER ( IC = NLON )
      PARAMETER ( IWK = 2*NLON*NLAT + 2*(NLON+NLAT) )
C
C

      INTEGER*2  I2,I3,I17,I18,I19,I20,I21,I22,I23,I24,I25,I26,I27,I28
      INTEGER*4  I1,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,I15,I16
      REAL*8     R1,R2
      REAL*4     R3,R4
      REAL*4     LON(NLON), LAT(NLAT), GRID(NLON,NLAT)
      REAL*4     C(2,NLON,2,NLAT), WK(IWK)
      REAL*4     GLAT, ELON, HSS, GUND
      REAL*4     DUR, SLAT1, SLAT2, SLON1, SLON2
```

```
      REAL*8      STIM1, STIM2
      INTEGER     ING, INP, OUP, OUR
      INTEGER     IREC, LPP, NLINE, NPAGE, NREC
      INTEGER     I1S, I6S, IDUM1
      CHARACTER   TITLE*20 /' BLACK SEA DATA    '/
      CHARACTER   NS*12 /' NORTH-SOUTH'/
      CHARACTER   SN*12 /' SOUTH-NORTH'/
      CHARACTER   EW*12 /'SINGLE-POINT'/
      CHARACTER   DIR*12
C
      DATA        ING / 5 /, INP / 8 /, OUP / 6 /, OUR / 9 /
      DATA        K75 / 0 /, K78 / 0 /
      DATA        LPP / 54 /, NLINE / 1 /, NPAGE / 1 /, NREC / 0 /
C
C
C.....GENERATE A GRID NET AND READ IN RAPP'S OSU300 180X180 GEOID
C
      LON(1) = XMIN
      DO 100  I = 2, NLON
        LON(I) = LON(I-1) + DX
  100 CONTINUE
      LAT(1) = YMIN
      DO 110  J = 2, NLAT
        LAT(J) = LAT(J-1) + DY
  110 CONTINUE
C
      DO 120  J = NLAT, 1, -1
        READ ( ING, * )  RL
        READ ( ING, 501 )  (GRID(I,J),I=1,NLON)
  120 CONTINUE
C
C.....DETERMINE A BICUBIC SPLINE TWO-DIMENSIONAL COEFFICIENT MATRIX
C
      CALL IBCCCU ( GRID, LON,NLON, LAT,NLAT, C, IC, WK, IER )
C
C
C.....START PROCESSING THE GEODYN DATA SET
C
      READ ( INP )  I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &              I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &              I20,I21,I22,I23,I24,I25,I26,I27,I28
      NREV = I14
      I1S = I1
      I6S = I6
      SLAT1 = 1.0E-6 * FLOAT(I7)
      SLON1 = 1.0E-6 * FLOAT(I8)
      STIM1 = R1
      IF ( I1. EQ. 7502701 ) K75 = K75 + 1
      IF ( I1. EQ. 7806401 ) K78 = K78 + 1
      WRITE ( OUP, 601 )  TITLE, I6, NREV, I1, NPAGE
      WRITE ( OUP, 602 )
      REWIND INP
C
      DO 111  MJ = 1, MAXREC
C
      READ ( INP, END = 2000 ) I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &                 I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &                 I20,I21,I22,I23,I24,I25,I26,I27,I28
      NREC = NREC + 1
      GLAT = 1.0E-6 * FLOAT(I7)
      ELON = 1.0E-6 * FLOAT(I8)
      HSS  = 1.0E-3 * FLOAT(I13)
C
      CALL IBCEVL ( LON,NLON, LAT,NLAT, C,IC, ELON,GLAT,GUND, IER )
C
      WRITE ( OUR, 660 )  GLAT, ELON, GUND
C
      IF ( I14. EQ. NREV )  THEN
        IF ( NLINE. LE. LPP )  THEN
          WRITE( OUP, 603 )  NREC,R1,R2,GLAT,ELON,HSS,R3,GUND
```

```
            NLINE = NLINE + 1
            IREC = IREC + 1
          ELSE
            NLINE = 1
            IREC = IREC + 1
            NPAGE = NPAGE + 1
            IF( (I6-I6S). EQ. 1 )  R1 = R1 + 1.0
            WRITE ( OUP, 601 )  TITLE, I6, NREV, I1, NPAGE
            WRITE ( OUP, 602 )
            WRITE ( OUP, 603 )  NREC,R1,R2,GLAT,ELON,HSS,R3,GUND
          END IF
          SLAT2 = GLAT
          SLON2 = ELON
          STIM2 = R1
        ELSE
          IF ( SLAT2. EQ. 0.0. AND. SLON2. EQ. 0.0 )  THEN
            DIR = EW
            DUR = 0.0
          ELSE
            IF ( (SLAT2-SLAT1). LE. 0.0 )  THEN
              DIR = NS
            ELSE
              DIR = SN
            END IF
            DUR = 86400.0 * (STIM2 - STIM1 )
          END IF
          WRITE ( OUP, 610 )  DIR, DUR, IREC
          IF ( I1. EQ. 7502701 ) K75 = K75 + 1
          IF ( I1. EQ. 7806401 ) K78 = K78 + 1
          NREV = I14
          I1S = I1
          I6S = I6
          NLINE = 1
          IREC = 1
          NPAGE = NPAGE + 1
          WRITE ( OUP, 601 )  TITLE, I6, NREV, I1, NPAGE
          WRITE ( OUP, 602 )
          WRITE ( OUP, 603 )  NREC,R1,R2,GLAT,ELON,HSS,R3,GUND
          SLAT1 = GLAT
          SLON1 = ELON
          STIM1 = R1
          SLAT2 = 0.
          SLON2 = 0.
          STIM2 = 0.
        END IF
C
  111 CONTINUE
C
 2000 CONTINUE
C
        IF ( SLAT2. EQ. 0.0. AND. SLON2. EQ. 0.0 )  THEN
          DIR = EW
          DUR = 0.0
        ELSE
          IF ( (SLAT2-SLAT1). LE. 0.0 )  THEN
            DIR = NS
          ELSE
            DIR = SN
          END IF
          DUR = 86400.0 * (STIM2 - STIM1 )
        END IF
        WRITE ( OUP, 610 )  DIR, DUR, IREC
C
        WRITE( OUP, 611 )  K75, K78
C
  501 FORMAT((6(2X,F10.3)))
  601 FORMAT('1',T2,A20,8X,'MOD. JULIAN DAY = ',I6,8X,'PASS = ',I6,
     &           8X,'SATELLITE ID = ',I10,T120,'PAGE ',I4,/)
  602 FORMAT(' ',' RECORD # ',2X,'     FRACTION OF DAY',3X,
     &           '   ALTIMETER OBS. (M)',5X,'  S/C GEOD LAT ',
```

```
     &           ' S/C EAST LONG ',' ELEVATION (M)',5X,' RAPP-180 '/
     &       '+',',__________',2X,'_____________',3X,
     &           ',_____________',5X,'_____',5X,'_____')
  603 FORMAT(' ',I7,6X,F20.18,3X,F20.12,5X,2(4X,F9.6,2X),F9.3,
     &           '(',F4.2,')',5X,F10.3)
  604 FORMAT(I5,I10,2F8.3,4F10.6)
  610 FORMAT(1H ,/,T5,'THE APPROX. DURATION OF THIS ',A12,' PASS IS',
     &          F8.2,' SEC. WITH ',I6,' RECORDS.')
  611 FORMAT(1H1,T20,'# OF PASSES FOR SATELLITE GEOS-3 IS:',I6,//
     &           T20,'# OF PASSES FOR SATELLITE SEASAT IS:',I6)
  660 FORMAT(5X,2F9.3,45X,F9.3)
C
      STOP
      END
//*
// EXEC LINKGOV,REGION.GO=5000K
//SYSLIB   DD DSN=SYS2.IMSLS,DISP=SHR
//*
//GO.FT05F001 DD DSN=ZMAYA.ALTIM.DATA(BRH180),DISP=SHR
//GO.FT06F001 DD SYSOUT=*
//GO.FT08F001 DD DSN=ZMAYA.BLACK.DATA,DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** *
//*                                                        *
//*           BLACK SEA DATA                               *
//*                                                        *
//* **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT09F001 DD DSN=ZMAYA.ALTIM.DATA(BLKRAPP),DISP=SHR
//*
// EXEC NOTIFYTS
```

A listing of PROGRAM PLTGRP

```
//ZMAYABEP JOB (G0109,360,75),AYAU,TIME=(12,00),CLASS=O,MSGCLASS=X
/*JOBPARM LINES=60
//TEMPLATE EXEC G38PLOT
//FORT.SYSIN DD *
C
C FORMAT OF SEASAT ALTIMETER DATA
C VARIABLE    TYPE    DESCRIPTION
C              I4     SATELITE ID
C              I2     MEASUREMENT TYPE ( 42= OVER LAND , 43 = OVER WATER )
C              I2     TIME SYSTEM ( NM )
C              I4     STATION NUMBER
C              I4     PREPROCESSING INDICATORS
C      *       I4     MODIFIED JULIAN DATE OF OBSERVATION
C      *       R8     FRACTION OF DAY PAST MIDNIGHT  (GMT)
C      *       R8     ALTIMETER OBSERVATION          (METERS)
C      *       I4     SATELLITE GEODETIC LATITUDE    (1E-6 DEGREES)
C      *       I4     SATELLITE EAST LONGITUDE       (1E-6 DEGREES)
C              R4     MEASUREMENT STANDARD DEVIATION (METERS)
C              I4     NET INSTRUMENT CORRECTION      (MM)
C              I4     METEOROLOGICAL DATA WORD       (GEODYN VOL 3)
C              I4     NET MEDIA CORRECTIONS          (MM)
C              R4     GEOID HEIGHT ABOVE REFERENCE ELLIPSOID (METERS)
C              I4     NET OCEAN DYNAMIC CORRECTIONS  (MM)
C      *       I4     INDICATED SURFACE ELEVATION (MM)
C      *       I4     S/C REVOLUTION NUMBER
C              I4     MEAN SEA SURFACE ELEVATION (MARSH/MARTIN '81 (MM))
C              I4     DOD REFERENCE RADIAL ORBIT DIFFERENCE (MM)
C              I2     H 1/3                          (CM)
C              I2     AGC                            (DB)
C              I2     WIND SPEED                     (CM/SEC)
C              I2     SURFACE ELEVATION PREPROCESSING WORD
C              I2     DRY TROPOSPHERIC CORRECTION      (MM)
C              I2     FNOC WET TROPOSPHERIC CORRECTION (MM)
C              I2     SMMR WET TROPOSPHERIC CORRECTION (MM)
C              I2     IONOSPHERIC CORRECTION           (MM)
C              I2     BAROTROPIC DYNAMIC SEA SURFACE CORRECTION (MM)
C              I2     SOLID EARTH TIDE                  (MM)
C              I2     SCHWIDERSKI OCEAN TIDE            (MM)
C              I2     PARKE OCEAN TIDE                  (MM)
C
      INTEGER*2  I2,I3,I17,I18,I19,I20,I21,I22,I23,I24,I25,I26,I27,I28
      INTEGER*4  I1,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,I15,I16
      REAL*8     R1,R2
      REAL*4     R3,R4
      REAL*4     GLAT, ELON, HSS
      REAL*4     X(5000),Y(5000)
      INTEGER    INP, IREC, LPP, NLINE, NPAGE, IPLOT, NREC, NREV, OUP
      INTEGER    IWRT, MAXPLT, LIST(6)
      CHARACTER  TITLE*20 /' BLACK SEA DATA     '/
C
      DATA   INP / 5 /, OUP / 6 /
      DATA   LIST / 8, 110, 75, 0, 7, 10 /
      DATA   LPP / 54 /, NLINE / 1 /, NPAGE / 1 /
      DATA   IREC / 0 /, NREC / 0 /, IPLOT / 0 /
      DATA   IWRT / 0 /, MAXPLT / 99999 /
C
      READ ( INP )  I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &              I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &              I20,I21,I22,I23,I24,I25,I26,I27,I28
      NREV = I14
      IF ( IWRT. GT. 0 )  THEN
        WRITE ( OUP, 601 )  TITLE, NREV, NPAGE
        WRITE ( OUP, 602 )
      END IF
      REWIND INP
      CALL USTART
      CALL UINQES(1.0,SUPRT)
```

```
      IF(SUPRT.NE.0.0)  CALL UESCAP(1.0,LIST,6.0,IDUM,DUM)
C
 1000 READ ( INP, END = 2000 ) I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     *                         I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     *                         I20,I21,I22,I23,I24,I25,I26,I27,I28
      NREC = NREC + 1
      GLAT = 1.0E-6 * FLOAT(I7)
      ELON = 1.0E-6 * FLOAT(I8)
      HSS  = 1.0E-3 * FLOAT(I13)
C
      IF ( I14. EQ. NREV )  THEN
        IF ( NLINE. LE. LPP )  THEN
          IF (IWRT.GT.0)  WRITE(OUP, 603)  I6,R1,R2,GLAT,ELON,HSS,NREC
          NLINE = NLINE + 1
        ELSE
          NLINE = 1
          NPAGE = NPAGE + 1
          IF ( IWRT. GT. 0 )  THEN
            WRITE ( OUP, 601 )  TITLE, NREV, NPAGE
            WRITE ( OUP, 602 )
            WRITE ( OUP, 603 )  I6,R1,R2,GLAT,ELON,HSS,NREC
          END IF
        END IF
        IREC = IREC + 1
        RMJD = FLOAT(I6)
        RNREV = FLOAT(I14)
        X(IREC) = R1
        Y(IREC) = HSS
      ELSE
C
C.....PLOT THE PREVIOUS ELEVATION PROFILE
C
        IF ( IPLOT. GT. MAXPLT )  GO TO 3000
        IF ( IREC. GT. 3. AND. IPLOT. LE. MAXPLT )  THEN
C
          IPLOT = IPLOT + 1
          CALL PLOT(X,Y,IREC,RMJD,RNREV)
C
        END IF
C
        NREV = I14
        NLINE = 1
        NPAGE = NPAGE + 1
        IREC = 1
        X(IREC) = R1
        Y(IREC) = HSS
        IF ( IWRT. GT. 0 )  THEN
          WRITE ( OUP, 601 )  TITLE, NREV, NPAGE
          WRITE ( OUP, 602 )
          WRITE ( OUP, 603 )  I6,R1,R2,GLAT,ELON,HSS,NREC
        END IF
      END IF
C
      GO TO 1000
C
 2000 CONTINUE
C
      IF ( IREC. GT. 3. AND. IPLOT. LE. MAXPLT )  THEN
        IPLOT = IPLOT + 1
        CALL PLOT(X,Y,IREC,RMJD,RNREV)
      END IF
C
 3000 WRITE( OUP, * ) ' '
      WRITE( OUP, * ) ' ', NREC, ' RECORDS IN FILE.'
      WRITE( OUP, * ) ' '
      WRITE( OUP, * ) ' ', IPLOT, ' ELEVATION PROFILES PLOTTED.'
C
      CALL UEND
C
  601 FORMAT('1',T10,A20,10X,'PASS # = ',I6,T90,'PAGE ',I4,/)
```

```
  602 FORMAT(' ','JULIAN DAY',2X,'      FRACTION OF DAY',3X,
     &             '    ALTIMETER OBS. (M)',5X,'  S/C GEOD LAT ',
     &             ' S/C EAST LONG ',' ELEVATION (M) ',5X,' RECORD # '/
     &        '+','___________',2X,'      _________________',3X,
     &             '    _____________________',5X,'    ____________   ',
     &             '   ______________',5X,'     ___________            ')
  603 FORMAT(' ',I7,6X,F20.18,3X,F20.12,5X,2(4X,F9.6,2X),5X,F7.3,5X,I10)
C
      STOP
      END
C
      SUBROUTINE PLOT ( X, Y, IREC, RMJD, RNREV )
C
      REAL*4        X(IREC), Y(IREC), RMJD, RNREV
      INTEGER       IREC
      CHARACTER*1   DATE(10)
C
      CALL USET('PERCENTUNITS')
      CALL USET('EXTRALARGE')
      CALL UVWPRT(0.0, 99.0, 0.0, 99.0)
      CALL UOUTLN
C
C.....PRINT TITLE, JULIAN DATE AND PASS NUMBER.
C
      CALL USET('CJUST')
      CALL USET('TJUST')
      CALL UPSET('PRECISION',5.0)
      CALL UPRINT(50.,96.,
     &            'BLACK SEA ALTIMETER ELEVATION PROFILE$')
      CALL UPRINT(50.,92.,'MODIFIED JULIAN DATE : $')
      CALL UMOVE(70.,92.)
      CALL UPRNT1(RMJD,'REAL')
      CALL UPRINT(50.,88.,'PASS : $')
      CALL UMOVE(60.,88.)
      CALL UPRNT1(RNREV,'REAL')
      CALL ZTIME(DATE,8)
      CALL FMOVE(DATE(10),1,'$')
      CALL UPRINT(88.,6.,DATE)
      CALL UPRINT(88.,3.,'STX/ZMAYA$')
C
C.....DRAW AND LABEL AXES  (DEFAULT TIC MARKS)
C
      CALL USET('NOORIGIN')
      CALL UVWPRT(5.,95.,7.,90.)
      CALL USET('DSYMBOL')
      CALL UPSET('SYMBOL',5.0)
      CALL UPSET('SZSYMBOL',1.0)
      CALL USET('LARGE')
      CALL UPSET('XLABEL','FRACTION OF DAY PAST MIDNIGHT$')
      CALL UPSET('YLABEL','SURFACE ELEVATION IN METERS$')
      CALL USET('XBOTH')
      CALL USET('YBOTH')
      CALL USET('OWNSCALE')
C     CALL UPSET('TICX',TCX)
C     CALL UPSET('TICY',TCY)
      XMIN = X(1)
      XMAX = X(IREC)
      YMIN = Y(1)
      YMAX = Y(1)
      DO 111  I = 1,IREC
        YMIN = AMIN1 ( Y(I), YMIN )
        YMAX = AMAX1 ( Y(I), YMAX )
  111 CONTINUE
      SX = 0.05 * ( XMAX - XMIN )
      SY = 0.10 * ( YMAX - YMIN )
      XMIN = XMIN - SX
      XMAX = XMAX + SX
      YMIN = YMIN - SY
      YMAX = YMAX + SY
      CALL UWINDO(XMIN,XMAX,YMIN,YMAX)
```

```
      CALL UAXIS(XMIN,XMAX,YMIN,YMAX)
C
C.....PLOT POLY-LINE
C
      XPT = FLOAT(IREC)
      CALL ULINE(X,Y,XPT)
C
C.....DISPLAY FOR SCREEN
C
C     CALL UPAUSE
C
C.....TERMINATED THIS PLOT
C
      CALL UERASE
      CALL URESET
C
      RETURN
      END
//*
//GO.FT05F001 DD DSN=ZMAYA.BLACK.DATA,DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** *
//*                                                        *
//*                   BLACK SEA DATA                       *
//*                                                        *
//* **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT06F001 DD SYSOUT=*
// EXEC NOTIFYTS
```

A listing of PROGRAM GRNTRK

```
//ZMAYABAK JOB (G0109,360,7),AYAU,TIME=(1,00),CLASS=O,MSGCLASS=X
/*JOBPARM LINES=60
//TEMPLATE EXEC G38PLOT
//FORT.SYSIN DD *
C
C FORMAT OF GEOS-3/SEASAT ALTIMETER DATA
C VARIABLE    TYPE    DESCRIPTION
C              I4     SATELITE ID
C              I2     MEASUREMENT TYPE ( 42= OVER LAND , 43 = OVER WATER )
C              I2     TIME SYSTEM ( NM )
C              I4     STATION NUMBER
C              I4     PREPROCESSING INDICATORS
C      *       I4     MODIFIED JULIAN DATE OF OBSERVATION
C      *       R8     FRACTION OF DAY PAST MIDNIGHT  (GMT)
C      *       R8     ALTIMETER OBSERVATION          (METERS)
C      *       I4     SATELLITE GEODETIC LATITUDE    (1E-6 DEGREES)
C      *       I4     SATELLITE EAST LONGITUDE       (1E-6 DEGREES)
C              R4     MEASUREMENT STANDARD DEVIATION (METERS)
C              I4     NET INSTRUMENT CORRECTION      (MM)
C              I4     METEOROLOGICAL DATA WORD       (GEODYN VOL 3)
C              I4     NET MEDIA CORRECTIONS          (MM)
C              R4     GEOID HEIGHT ABOVE REFERENCE ELLIPSOID (METERS)
C              I4     NET OCEAN DYNAMIC CORRECTIONS  (MM)
C      *       I4     INDICATED SURFACE ELEVATION (MM)
C      *       I4     S/C REVOLUTION NUMBER
C              I4     MEAN SEA SURFACE ELEVATION (MARSH/MARTIN '81 (MM))
C              I4     DOD REFERENCE RADIAL ORBIT DIFFERENCE (MM)
C              I2     H 1/3                          (CM)
C              I2     AGC                            (DB)
C              I2     WIND SPEED                     (CM/SEC)
C              I2     SURFACE ELEVATION PREPROCESSING WORD
C              I2     DRY TROPOSPHERIC CORRECTION      (MM)
C              I2     FNOC WET TROPOSPHERIC CORRECTION (MM)
C              I2     SMMR WET TROPOSPHERIC CORRECTION (MM)
C              I2     IONOSPHERIC CORRECTION           (MM)
C              I2     BAROTROPIC DYNAMIC SEA SURFACE CORRECTION (MM)
C              I2     SOLID EARTH TIDE                 (MM)
C              I2     SCHWIDERSKI OCEAN TIDE           (MM)
C              I2     PARKE OCEAN TIDE                 (MM)
C
      PARAMETER ( NMAX = 1000 )
      PARAMETER ( MAXREC = 10000 )
C
      INTEGER*2  I2,I3,I17,I18,I19,I20,I21,I22,I23,I24,I25,I26,I27,I28
      INTEGER*4  I1,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,I15,I16
      INTEGER    IGRID, INP, IREC, ITRK, LPP, NLINE, NPAGE, NREC, OUP
      INTEGER    K75, K78
      REAL*8     R1,R2
      REAL*4     R3,R4
      REAL*4     GLAT, ELON, HSS
      REAL*4     X(NMAX),Y(NMAX), XP(NMAX),YP(NMAX)
C
      INTEGER      NPROJ, LINEH, LINEV
      INTEGER      LIST(6)
      REAL*4       PENS(7)
      REAL*4       PLAT,PLONG,VLAT1,VLAT2,VLONG1,VLONG2
      CHARACTER*1  DATE(10)
C
      DATA  PENS / 7*4.0 /
      DATA  LIST / 8, 110, 75, 0, 7, 10 /
      DATA  NPROJ / 16 /
      DATA  PLAT / 90. /, PLONG / 0. /
      DATA  VLONG1 / 26.5 /, VLONG2 / 42.5 /
      DATA  VLAT1 / 40. /, VLAT2 / 48. /
      DATA  LINEH / 0 /, LINEV / 0 /
C
      DATA  INP / 5 /, OUP / 6 /
```

```
      DATA  IREC / 0 /, NREC / 0 /, ITRK / 0 /
      DATA  IGRID / 1 /
C
C.....PROJECT REGIONAL MAP
C
      CALL USTART
      CALL UINQES(1.0,SUPRT)
      IF (SUPRT.NE.0.0)  CALL UESCAP(1.0,LIST,6.0,IDUM,DUM)
      CALL USET('PERCENTUNITS')
      CALL UVWPRT(0.0,99.0,0.0,99.0)
      CALL UOUTLN
C
C.....PUT ON LABELS
C
      CALL USET('CJUST')
      CALL USET('TJUST')
      CALL USET('YELLOW')
      CALL USET('SOFTWARE')
      CALL USET('EXTRALARGE')
      CALL UPSET('VERTICAL SIZE',2.0)
      CALL UPSET('HORIZONT SIZE',1.6)
      CALL UFONT('TROM')
      CALL UPRINT(50.,96.,
     &            'GEOS-3/SEASAT GROUND TRACKS OVER BLACK SEA$')
      IF ( IGRID. GT. 0 )  THEN
        LINEH = IFIX( VLAT2 - VLAT1 )
        LINEV = IFIX( VLONG2 - VLONG1 )
      END IF
      CALL WDMAP(NPROJ,PLAT,PLONG,VLAT1,VLAT2,VLONG1,VLONG2,
     &  LINEH,LINEV)
C
C.....LABEL AXIS AND WRITE LON AND LATS ON PLOT
C
      IF ( IGRID. GT. 0 )  THEN
C
        DLONG = ABS(VLONG2 - VLONG1) / LINEV
        LINES = LINEV + 1
        DIX = 100.0 / LINEV
        YLONGI = -3.5
        DO 420 I = 1, LINES, 2
          V = VLONG1 + (DLONG * (I - 1))
          XLATIT = DIX * (I - 1)
          CALL UMOVE(XLATIT,YLONGI)
          CALL UPRNT1(V,'REAL')
  420   CONTINUE
C
C.....WRITE LABEL
C
        CALL USET('BJUS')
        CALL UMOVE(50.,-7.5)
        CALL UPRNT1('EAST LONGITUDES$','HORIZ')
C
        DLAT = ABS(VLAT2 - VLAT1) / LINEH
        LINES = LINEH + 1
        DIY = 100.0 / LINEH
        XLT = -3.5
        DO 430 I = 1, LINES, 2
          V = VLAT1 + (DLAT * (I - 1))
          YLO = DIY * (I - 1)
          CALL UMOVE(XLT,YLO)
          CALL UPRNT1(V,'REAL')
  430   CONTINUE
C
C.....WRITE LABEL
C
        CALL USET('MJUS')
        CALL UMOVE(-7.5,50.0)
        CALL UPRNT1('GEODETIC LATITUDES$','VERTI')
C
      END IF
```

```
C
C.....WRITE DATE AND ID ON PLOT
C
      CALL ZTIME(DATE,8)
      CALL FMOVE(DATE(10),1,'$')
      CALL UPRINT(79.,-8.,DATE)
      CALL UPRINT(96.,-8.,'STX/ZMAYA$')
C
      READ ( INP )  I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &              I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &              I20,I21,I22,I23,I24,I25,I26,I27,I28
      NREV = I14
      IF ( I1. EQ. 7502701 )  K75 = K75 + 1
      IF ( I1. EQ. 7806401 )  K78 = K78 + 1
      REWIND INP
C
      DO 111 IJK = 1, MAXREC
C
      READ ( INP, END = 2000 ) I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &                 I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &                 I20,I21,I22,I23,I24,I25,I26,I27,I28
      NREC = NREC + 1
      GLAT = 1.0E-6 * FLOAT(I7)
      ELON = 1.0E-6 * FLOAT(I8)
      HSS  = 1.0E-3 * FLOAT(I13)
C
      IF ( I14. EQ. NREV )  THEN
        IREC = IREC + 1
        X(IREC) = ELON
        Y(IREC) = GLAT
      ELSE
C
C.....SET UP TO PLOT THE PREVIOUS GROUND TRACK
C
        IF ( IREC. GT. 1 )  THEN
C
          ITRK = ITRK + 1
          CALL PLGRTK (X,Y, XP,YP, IREC)
C
        END IF
C
        NREV = I14
        IF ( I1. EQ. 7502701 )  K75 = K75 + 1
        IF ( I1. EQ. 7806401 )  K78 = K78 + 1
        IREC = 1
        X(IREC) = ELON
        Y(IREC) = GLAT
      END IF
C
  111 CONTINUE
C
 2000 CONTINUE
C
      IF ( IREC. GT. 1 )  THEN
        ITRK = ITRK + 1
        CALL PLGRTK(X,Y,XP,YP,IREC)
      END IF
C
      WRITE( OUP, * ) ' '
      WRITE( OUP, * ) ' ', NREC, ' RECORDS IN FILE.'
      WRITE( OUP, * ) ' '
      WRITE( OUP, * ) ' ', ITRK, ' GROUND TRACKS PLOTTED, WHERE'
      WRITE( OUP, * ) ' '
      WRITE( OUP, * ) '     ', K75, ' TRACKS FROM GEOS-3 AND'
      WRITE( OUP, * ) ' '
      WRITE( OUP, * ) '     ', K78, ' TRACKS FROM SEASAT.'
C
C     CALL UPAUSE
      CALL UEND
C
```

```
      STOP
      END
C
      SUBROUTINE PLGRTK ( X, Y, XP, YP, IREC )
C
      REAL*4  X(IREC), Y(IREC), XP(IREC), YP(IREC)
      INTEGER IREC
C
      CALL USET('PERCENTUNITS')
      CALL UVWPRT(0.0, 99.0, 0.0, 99.0)
      CALL USET('OWNSCALE')
C
C.....PLOT GROUND TRACK
C
      DO 111  I = 1, IREC
        CALL WDPOS(Y(I),X(I),XP(I),YP(I))
  111 CONTINUE
C
      CALL USET('LNULL')
      XPT = FLOAT(IREC)
      CALL ULINE(XP,YP,XPT)
C
      RETURN
      END
//*
//GO.FT05F001 DD DSN=ZMAYA.BLACKA.DATA,DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** **** *
//*                                                             *
//*                    BLACK SEA DATA                           *
//*                                                             *
//* **** **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT06F001 DD SYSOUT=*
//GO.FT19F001  DD DSN=SYS2.WRLDATA2,DISP=SHR,LABEL=(,,,IN)
// EXEC NOTIFYTS
```

A listing of PROGRAM XOVER0

```
//ZMAYABXO JOB (G0109,360,2),AYAU,TIME=(0,30),CLASS=O,MSGCLASS=X
// EXEC FORTVC
//SYSIN DD *
C
C FORMAT OF GEOS-3/SEASAT ALTIMETER DATA
C
C VARIABLE    TYPE    DESCRIPTION
C
C     *        I4     SATELITE ID
C              I2     MEASUREMENT TYPE ( 42= OVER LAND , 43 = OVER WATER )
C              I2     TIME SYSTEM ( NM )
C              I4     STATION NUMBER
C              I4     PREPROCESSING INDICATORS
C              I4     MODIFIED JULIAN DATE OF OBSERVATION
C              R8     FRACTION OF DAY PAST MIDNIGHT     (GMT)
C              R8     ALTIMETER OBSERVATION             (METERS)
C     *        I4     SATELLITE GEODETIC LATITUDE       (1E-6 DEGREES)
C     *        I4     SATELLITE EAST LONGITUDE          (1E-6 DEGREES)
C              R4     MEASUREMENT STANDARD DEVIATION    (METERS)
C              I4     NET INSTRUMENT CORRECTION         (MM)
C              I4     METEOROLOGICAL DATA WORD          (GEODYN VOL 3)
C              I4     NET MEDIA CORRECTIONS             (MM)
C              R4     GEOID HEIGHT ABOVE REFERENCE ELLIPSOID (METERS)
C              I4     NET OCEAN DYNAMIC CORRECTIONS     (MM)
C              I4     INDICATED SURFACE ELEVATION       (MM)
C     *        I4     S/C REVOLUTION NUMBER
C              I4     MEAN SEA SURFACE ELEVATION (MARSH/MARTIN '81 (MM))
C              I4     DOD REFERENCE RADIAL ORBIT DIFFERENCE (MM)
C              I2     H 1/3                             (CM)
C              I2     AGC                               (DB)
C              I2     WIND SPEED                        (CM/SEC)
C              I2     SURFACE ELEVATION PREPROCESSING WORD
C              I2     DRY TROPOSPHERIC CORRECTION       (MM)
C              I2     FNOC WET TROPOSPHERIC CORRECTION  (MM)
C              I2     SMMR WET TROPOSPHERIC CORRECTION  (MM)
C              I2     IONOSPHERIC CORRECTION            (MM)
C              I2     BAROTROPIC DYNAMIC SEA SURFACE CORRECTION (MM)
C              I2     SOLID EARTH TIDE                  (MM)
C              I2     SCHWIDERSKI OCEAN TIDE            (MM)
C              I2     PARKE OCEAN TIDE                  (MM)
C
      PARAMETER ( MPASS = 200 )
      PARAMETER ( NMAX =  200 )
      PARAMETER ( SIGE =  2.0 )
C
      INTEGER*2  I2,I3,I17,I18,I19,I20,I21,I22,I23,I24,I25,I26,I27,I28
      INTEGER*4  I1,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,I15,I16
      REAL*8     R1,R2
      REAL*4     R3,R4
      INTEGER*4  IDSAT
      REAL*4     GLAT, ELON
      REAL*4     X(NMAX),Y(NMAX), Z(NMAX), YR(NMAX)
      REAL*4     GX(NMAX),GY(NMAX), DDZ(NMAX), D(NMAX)
      REAL*8     F(NMAX,3), FT(3,NMAX)
      REAL*8     G(NMAX)
      REAL*8     A(MPASS), B(MPASS), C(MPASS)
      REAL*4     SLON1(MPASS), SLAT1(MPASS), SLON2(MPASS), SLAT2(MPASS)
      INTEGER*4  IDS(MPASS), NREV(MPASS), NBAD(MPASS)
      INTEGER*4  ICX(MPASS,MPASS), ISUM(MPASS), LN(MPASS), LY(MPASS)
      INTEGER    IDP, INP, IREC, NPASS, NREC, OUP
C
      DATA   INP / 3 /, OUP / 6 /
      DATA   IREC / 0 /, NREC / 0 /, NPASS / 0 /
C
      READ ( INP )  I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &              I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &              I20,I21,I22,I23,I24,I25,I26,I27,I28
```

```
      IDP = I14
      REWIND INP
C
      WRITE ( OUP, * ) ' ','     CURVE FITTING FOR EACH SATELLITE PASS'
      WRITE ( OUP, * ) ' ','
C
 1000 READ ( INP, END = 2000 ) I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &                 I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &                 I20,I21,I22,I23,I24,I25,I26,I27,I28
C*****
      IF ( I14. EQ. 12793 )  GO TO 999
C*****
      NREC = NREC + 1
      GLAT = 1.0E-6 * FLOAT(I7)
      ELON = 1.0E-6 * FLOAT(I8)
      HSS  = 1.0E-3 * FLOAT(I13)
      IDSAT = I1
C
      IF ( I14. EQ. IDP )  THEN
        IREC = IREC + 1
        IDP = I14
        X(IREC) = ELON
        Y(IREC) = GLAT
        Z(IREC) = HSS
      ELSE
C
C.....DETERMINE THE SECOND-ORDER POLYNOMIAL FOR A PASS
C
        IF ( IREC. GE. 5 )    THEN
C
          NPASS = NPASS + 1
          NREV(NPASS) = IDP
          IDS(NPASS) = IDSAT
          SLON1(NPASS) = X(1)
          SLAT1(NPASS) = Y(1)
          SLON2(NPASS) = X(IREC)
          SLAT2(NPASS) = Y(IREC)
          CALL CURVE ( NPASS, X, Y, Z, YR, IREC, NREV(NPASS),
     &                 IDS(NPASS), A(NPASS), B(NPASS), C(NPASS),
     &                 NBAD(NPASS), F, FT, G, GX, GY, DDZ, D, SIGE )
C
        END IF
C
        IDP = I14
        IREC = 1
        X(IREC) = ELON
        Y(IREC) = GLAT
        Z(IREC) = HSS
      END IF
C
  999 CONTINUE
      GO TO 1000
C
 2000 CONTINUE
C
      IF ( IREC. GE. 5 )    THEN
C
        NPASS = NPASS + 1
        NREV(NPASS) = IDP
        IDS(NPASS) = IDSAT
        SLON1(NPASS) = X(1)
        SLAT1(NPASS) = Y(1)
        SLON2(NPASS) = X(IREC)
        SLAT2(NPASS) = Y(IREC)
        CALL CURVE ( NPASS, X, Y, Z, YR, IREC, NREV(NPASS),
     &               IDS(NPASS), A(NPASS), B(NPASS), C(NPASS),
     &               NBAD(NPASS), F, FT, G, GX, GY, DDZ, D, SIGE)
C
      END IF
C
```

```
 3000 WRITE( OUP, * ) ' '
      WRITE( OUP, * ) ' ', NREC, ' RECORDS IN FILE.'
      WRITE( OUP, * ) ' '
      WRITE( OUP, * ) ' ', NPASS, ' PASSES REPRESENTED BY AX**2+BX+C'
C
      CALL XOVER ( NPASS, NREV, IDS, A, B, C, SLON1, SLAT1,
     &             SLON2, SLAT2, ICX, ISUM, LN, LY, NBAD )
C
      STOP
      END
C
      SUBROUTINE CURVE ( NPASS, X, Y, Z, YR, IREC, IDP, IDSAT, A, B, C,
     &                   NB, F, FT, G, GX, GY, DDZ, D, SIGE )
C
      REAL*4        X(IREC), Y(IREC), Z(IREC), YR(IREC)
      REAL*4        GX(IREC), GY(IREC), DDZ(IREC), D(IREC)
      REAL*8        F(IREC,3), FT(3,IREC)
      REAL*8        E(3,3), EI(3,3), AE(6)
      REAL*8        G(IREC), H(3)
      REAL*8        A, B, C, DET, YD
      INTEGER*4     IDP, IDSAT, IREC, NB, NPASS
      INTEGER*4     IDIAG(3) / 1, 3, 6 /
      INTEGER*4     OUP / 6 /
C
C.....EVICT THE BAD DATA POINTS BY COMPARING THE ABSOLUTE DIFFERENCE
C       BETWEEN ADJACENT DATA POINTS
C
      NB = 0
      IREC1 = IREC - 1
C
      DO 111  I = 1, IREC
        GX(I) = -1.0
  111 CONTINUE
C
      DO 112  I = 1, IREC1
        DDZ(I) = ABS( Z(I+1) - Z(I) )
  112 CONTINUE
C
      DO 113  J = 1, IREC1
        IF(DDZ(J).LT.SIGE)  THEN
          K = J
          GO TO 200
        END IF
  113 CONTINUE
C
  200 CONTINUE
      DO 222  J = 1, IREC
        IF(DDZ(IREC-J).LT.SIGE)  THEN
          IRECG = IREC-J
          GO TO 300
        END IF
  222 CONTINUE
C
  300 KB = 0
      DO 333  I = 1, IRECG-1
        IF(DDZ(I).GT.SIGE)  KB = KB + 1
  333 CONTINUE
C
      LG = 1
      GX(LG) = X(K)
      GY(LG) = Y(K)
C
  400 LG = LG + 1
      K = K + 1
      IF(DDZ(K-1).GT.SIGE)  THEN
        K = K + 1
        DO 444  I = 1, KB-1
          IF(DDZ(K-1).GT.SIGE)  K = K + 1
  444   CONTINUE
      END IF
```

```
      GX(LG) = X(K)
      GY(LG) = Y(K)
      IF(K.LE.IRECG)  GO TO 400
C
      DO 555  I = 1, LG
        F(I,1) = GX(I)*GX(I)
        F(I,2) = GX(I)
        F(I,3) = 1.0D0
        G(I) = GY(I)
  555 CONTINUE
C
      DO 666  I = 1, LG
        DO 667  J = 1, 3
          FT(J,I) = F(I,J)
  667   CONTINUE
  666 CONTINUE
C
      DO 777  I = 1, 3
        DO 778  J = 1, 3
          E(I,J) = 0.D0
          DO 779  K = 1, LG
            E(I,J) = E(I,J) + FT(I,K)*F(K,J)
  779     CONTINUE
  778   CONTINUE
  777 CONTINUE
C
      DO 888  I = 1, 3
        H(I) = 0.D0
        DO 889  K = 1, LG
          H(I) = H(I) + FT(I,K)*G(K)
  889   CONTINUE
  888 CONTINUE
C
      AE(1) = E(1,1)
      AE(2) = E(1,2)
      AE(3) = E(2,2)
      AE(4) = E(1,3)
      AE(5) = E(2,3)
      AE(6) = E(3,3)
C
      CALL DSOLVE ( AE, H, IDIAG, 3, .TRUE., .TRUE. )
C
      A = H(1)
      B = H(2)
      C = H(3)
C
C     DET = E(1,1) * ( E(2,2)*E(3,3) - E(2,3)*E(2,3) ) +
C    &      E(1,2) * ( E(1,3)*E(2,3) - E(1,2)*E(3,3) ) +
C    &      E(1,3) * ( E(1,2)*E(2,3) - E(1,3)*E(2,2) )
C
C     EI(1,1) = ( E(2,2)*E(3,3) - E(2,3)*E(2,3) ) / DET
C     EI(2,2) = ( E(1,1)*E(3,3) - E(1,3)*E(1,3) ) / DET
C     EI(3,3) = ( E(1,1)*E(2,2) - E(1,2)*E(1,2) ) / DET
C     EI(2,3) = ( E(1,2)*E(1,3) - E(1,1)*E(2,3) ) / DET
C     EI(1,3) = ( E(1,2)*E(2,3) - E(1,3)*E(2,2) ) / DET
C     EI(1,2) = ( E(1,3)*E(2,3) - E(1,2)*E(3,3) ) / DET
C
C     A = ( EI(1,1)*H(1) + EI(1,2)*H(2) + EI(1,3)*H(3) )
C     B = ( EI(1,2)*H(1) + EI(2,2)*H(2) + EI(2,3)*H(3) )
C     C = ( EI(1,3)*H(1) + EI(2,3)*H(2) + EI(3,3)*H(3) )
C
      RMS = 0.
      DO 999  I = 1, LG
        YD = A*GX(I)*GX(I) + B*GX(I) + C
        YR(I) = SNGL(YD)
        D(I) = YR(I) - GY(I)
        RMS = RMS + D(I)*D(I)
  999 CONTINUE
      RMS = SQRT ( RMS/FLOAT(LG) )
C
```

```
      IF ( RMS. GT. 1.00 )  NB = 1
C
      WRITE ( OUP, 601 )  NPASS, IDP, IDSAT, A, B, C,
     &                    GX(1), GY(1), GX(LG), GY(LG), RMS, LG, NB
      IF ( RMS. GT. 0.005 )  THEN
        WRITE ( 6, * )  ' '
        WRITE ( 6, 602 )  (M,GX(M),GY(M),YR(M),D(M),M=1,LG)
        WRITE ( 6, * )  ' '
      END IF
C
  601 FORMAT(1H0,//,1X,I3,I7,I9,3E14.5,5F10.4,I6,I3)
  602 FORMAT((1H ,T5,I4,3X,3F10.4,2X,F10.4))
C
      RETURN
      END
C
      SUBROUTINE XOVER ( NPASS, NREV, IDSAT, A, B, C, SLON1, SLAT1,
     &                   SLON2, SLAT2, ICX, ISUM, LN, LY, NBAD )
C
      INTEGER    NREV(NPASS), IDSAT(NPASS), NBAD(NPASS)
      INTEGER    ICX(NPASS,NPASS), ISUM(NPASS), LN(NPASS), LY(NPASS)
      INTEGER    INDEX(2)
      REAL*4     SLAT1(NPASS), SLAT2(NPASS), SLON1(NPASS), SLON2(NPASS)
      REAL*8     A(NPASS), B(NPASS), C(NPASS)
      REAL*8     T(2)
      REAL*8     DA,DB,DC,RT,SRT,P,Q
      INTEGER    ITOTAL, IXM, MAXIX, MX, MXS, MX75, MX78, NPASS, OUP
      DATA       INP / 5 /,  OUP / 6 /
      DATA       MAXIX / 0 /, MX / 0 /, MX75 / 0 /, MX78 / 0 /
      DATA       ITOTAL / 0 /
C
      DO 123  I = 1, NPASS
        ISUM(I) = 0
        DO 234  J = 1, NPASS
          ICX(I,J) = 0
  234   CONTINUE
  123 CONTINUE
C
      WRITE ( OUP, 600 )
C
      DO 222  I = 1, NPASS-1
C
        IF ( NBAD(I). EQ. 0 )  THEN
C
          WRITE( OUP, 601 )  NREV(I), IDSAT(I), SLON1(I), SLAT1(I),
     &                       SLON2(I), SLAT2(I)
C
          DO 333  J = I+1, NPASS
C
            IF ( NBAD(J). EQ. 0 )  THEN
C
              INDEX(1) = 0
              INDEX(2) = 0
              DB = B(J) - B(I)
              DA =  A(I) - A(J)
              DC =  C(I) - C(J)
              RT = DB*DB - 4.0*DA*DC
C
              IF ( RT. GE. 0.0 )  THEN
                IF ( RT. EQ. 0.0 )  THEN
                  SRT = 0.0
                ELSE
                  SRT = DSQRT(RT)
                END IF
                T(1) = (DB+SRT)/(2.0*DA)
                T(2) = (DB-SRT)/(2.0*DA)
C               WRITE(6,*)  T, RT, SRT
C
                DO 135  L = 1, 2
                  IF ( SLON2(I). GT. SLON1(I) ) THEN
```

```
                IF( T(L). GE. SLON1(I). AND. T(L). LE. SLON2(I) )
     &            INDEX(L) = INDEX(L) + 1
                ELSE
                  IF( T(L). LE. SLON1(I). AND. T(L). GE. SLON2(I) )
     &              INDEX(L) = INDEX(L) + 1
                END IF
                IF ( SLON2(J). GT. SLON1(J) ) THEN
                  IF( T(L). GE. SLON1(J). AND. T(L). LE. SLON2(J) )
     &              INDEX(L) = INDEX(L) + 1
                ELSE
                  IF( T(L). LE. SLON1(J). AND. T(L). GE. SLON2(J) )
     &              INDEX(L) = INDEX(L) + 1
                END IF
  135           CONTINUE
C
                DO 246  L = 1, 2
                  IF ( INDEX(L). EQ. 2 )  THEN
                    MX = MX + 1
                    P = T(L)
                    Q = A(I)*P*P + B(I)*P + C(I)
                    WRITE ( OUP, 602)  NREV(J), IDSAT(J), P, Q,
     &                                 SLON1(J),SLAT1(J),
     &                                 SLON2(J),SLAT2(J)
                    ICX(I,J) = ICX(I,J) + 1
                    ICX(J,I) = ICX(I,J)
                    IF ( IDSAT(I). EQ. IDSAT(J) )  THEN
                      IF ( IDSAT(I). EQ. 7502701 )  MX75 = MX75 + 1
                      IF ( IDSAT(I). EQ. 7806401 )  MX78 = MX78 + 1
                    END IF
                  END IF
  246           CONTINUE
C
              END IF
C
            END IF
C
  333     CONTINUE
C
        END IF
C
  222 CONTINUE
C
      WRITE ( OUP, * )  ' '
      WRITE ( OUP, * )  ' '
      WRITE ( OUP, * )  '  TOTAL # OF POSSIBLE CROSSOVERS = ', MX
      WRITE ( OUP, * )  ' '
      WRITE ( OUP, * )  '  TOTAL # OF GEO3-GEO3 CROSSOVERS = ', MX75
      WRITE ( OUP, * )  ' '
      WRITE ( OUP, * )  '  TOTAL # OF SEASAT-SEASAT CROSSOVERS = ', MX78
      MXS = MX - MX75 - MX78
      WRITE ( OUP, * )  ' '
      WRITE ( OUP, * )  '  TOTAL # OF GEOS3-SEASAT CROSSOVERS = ', MXS
C
      DO 345  I = 1, NPASS
        DO 456  J = 1, NPASS
          ISUM(I) = ISUM(I) + ICX(I,J)
  456   CONTINUE
        ITOTAL = ITOTAL + ISUM(I)
  345 CONTINUE
      ITOTAL = ITOTAL/2
C
      DO 567  I = 1, NPASS
        MAXIX = MAX0 ( MAXIX, ISUM(I) )
  567 CONTINUE
C
      DO 678  I = 1, NPASS
        IF ( ISUM(I). EQ. MAXIX )  IXM = I
  678 CONTINUE
C
      WRITE ( OUP, 603 )
```

```
      WRITE ( OUP, 604 )  ( I, NREV(I), I = 1, NPASS )
      WRITE ( OUP, 605 )  NPASS
      WRITE ( OUP, 606 )  ( NREV(I),(ICX(I,J),J=1,94),ISUM(I),
     &                      I=1,NPASS )
      WRITE ( OUP, 607 )  NREV(IXM), MAXIX, ITOTAL
C
      MM = 0
      NN = 0
      DO 789  I = 1, NPASS
        IF ( ICX(IXM,I). EQ. 1 )  THEN
          MM = MM + 1
          LY(MM) = I
        ELSE
          NN = NN + 1
          LN(NN) = I
        END IF
  789 CONTINUE
C
      WRITE ( OUP, 608 )  NREV(IXM), MM, NN, MM+NN
      WRITE ( OUP, 609 )  ( NREV(LY(I)), I = 1, MM )
C
      DO 987  J = 1, NN
        WRITE ( OUP, 610 )  NREV(LN(J))
        DO 876  I = 1, NPASS
          IF ( ICX(I,LN(J)). EQ. 1 )  THEN
            WRITE ( OUP, 611 )  NREV(I)
            GO TO 888
          END IF
  876     CONTINUE
  888   CONTINUE
  987 CONTINUE
C
      WRITE ( 6, * ) ' '
      WRITE ( 6, 612 )
      DO 999  I = 1, NPASS
        DO 998  J = 1, NPASS
          IF ( ICX(I,J). GT. 1 )  WRITE ( 6, * )  NREV(I),NREV(J)
  998   CONTINUE
  999 CONTINUE
C
  501 FORMAT(I5,I10,2F8.3,4F10.6)
  600 FORMAT(1H1,T20,'BLACK SEA CROSSOVER LISTING:',//)
  601 FORMAT(1H0,//,T5,'PASS = ',I6,5X,'SAT ID = ',I10,5X,4F10.6//)
  602 FORMAT(1H ,T10,'XPASS = ',I6,5X,'SAT ID = ',I10,5X,2F10.3,
     &       5X,4F10.6)
  603 FORMAT(1H1,T20,'ROW-COLUMN ELEMENTS OF PASS CORRELATION MATRIX:',
     &       //)
  604 FORMAT((1H ,10(3X,I4,I6)))
  605 FORMAT(1H1,T20,'PARTIAL (94 COLUMNS) PASS CORRELATION MATRIX ',
     &                'FOR',I5,' PASSES:',//)
  606 FORMAT((1H ,10X,I8,5X,94I1,5X,I3))
  607 FORMAT(1H0,//,T10,'PASS # ',I6,' HAS THE MAXIUM OF ',I5,
     &       ' CROSSOVERS.',//,T10,'THE TOTAL # OF CROSSOVERS IS:',I5)
  608 FORMAT(1H1,T5,'REFERENCE PASS = ',I6,T110,3I6,//,
     &          T10,'THE DIRECT CROSS-PASSES ARE:',//)
  609 FORMAT((1H0,20I6))
  610 FORMAT(1H0,//,T5,'PASS: ',I6,'  IS INDIRECTLY RELATED TO THE')
  611 FORMAT(1H ,T10,'REFERENCE PASS VIA PASS ',I6)
  612 FORMAT(1H1,T10,'POSSIBLE COMBINATIONS OF PASSES THAT YIELD ',
     &                'BAD CROSSOVERS:',//)
C
      RETURN
      END
C
      SUBROUTINE DSOLVE (A,B, IDIAG,NEQ, FACT,BACK)
C
C  Compute the U**T * D * U  factorization of the symmetric matrix
C  stored in A, if FACT = TRUE;  and solve A * X = B if BACK = TRUE.
C
C
```

```
C  A          Contains the compacted-column form of the upper triangular
C             part of the coefficient matrix.  After factorization, it
C             contains D and U.
C  B          Right-hand-side vector.  After backsubstitution, it
C             contains the solution.
C
C  IDIAG      Addresses of the diagonal terms in A.
C  NEQ        Number of equations
C
C  FACT       If FACT = TRUE, then factor A;  otherwise do not factor A.
C  BACK       If BACK = TRUE, reduce B and backsubstitute;  otherwise
C             do not solve the equations.
C
C
C
      IMPLICIT REAL*8  (A-H,O-Z)
      LOGICAL          FACT,BACK
      DIMENSION        A(1),B(1),IDIAG(1)
C
C
C  Factor A, reduce B
      JR = 0
      DO 400  J = 1, NEQ
        JD = IDIAG(J)
        JH = JD - JR
        IS = J - JH + 2
C
        IF (JH .LT. 2)  GOTO 390
C
        IF (FACT) THEN
C
          IF (JH .GT. 2) THEN
C..       Reduce column J rows IS to J-1: do not divide by row diagonal
            K = JR + 2
            ID = IDIAG(IS - 1)
C
            DO 100 I = IS, J-1
              IR = ID
              ID = IDIAG(I)
              IH = MIN (ID-IR-1,I-IS+1)
              IF (IH.GT.0)  A(K) = A(K) - DOT(A(K-IH),A(ID-IH),IH)
              K = K + 1
  100       CONTINUE
          ENDIF
C
C..       Divide by row diagonal, and reduce diagonal term in column J
C
          IR = JR + 1
          K = J - JD
          DO 200  I = IR, JD-1
            ID = IDIAG(K+I)
            IF (A(ID).EQ.0.0) GOTO 200
            D = -A(I)
            A(I) = A(I)/A(ID)
            A(JD) = A(JD) + D*A(I)
  200     CONTINUE
C
        ENDIF
C
C..     Reduce RHS
        IF (BACK)  B(J) = B(J) - DOT(A(JR+1),B(IS-1),JH-1)
C
  390   JR = JD
  400 CONTINUE
C
C
      IF (.NOT.BACK)  RETURN
C
C
C  Divide by diagonal pivots
```

```
C
      DO 700 I = 1,NEQ
        ID = IDIAG(I)
        IF (A(ID).NE.0.0)    B(I) = B(I)/A(ID)
  700 CONTINUE
C
C  Backsubstitute
C
      J = NEQ
      JD = IDIAG(J)
C
  801 D = -B(J)
      J = J - 1
      IF (J.LE.0)  RETURN
C
      JR = IDIAG(J)
      IF (JD-JR.GT.1) THEN
        IS = J - JD + JR + 2
        K = JR - IS + 1
        DO 810  I = IS, J
  810   B(I) = B(I) + A(I+K)*D
      ENDIF
C
      JD = JR
      GOTO 801
C
      END
C
      FUNCTION DOT (A,B,N)
C
C  Compute the dot product of the two N-vectors A and B.
C
      INTEGER  N, I
      REAL*8   DOT, A(1),B(1)
C
      DOT=0.0
C
      DO 100  I=1,N
  100 DOT=DOT + A(I)*B(I)
C
      RETURN
      END
//*
// EXEC LINKGOV,REGION=3000K
//*SYSLIB    DD DSN=SYS2.IMSLD,DISP=SHR
//*          DD DSN=SYS2.IMSLS,DISP=SHR
//*
//GO.FT03F001 DD DSN=ZMAYA.BLACK.DATA,DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** *
//*                                                        *
//*                    BLACK SEA DATA                      *
//*                                                        *
//* **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT06F001 DD SYSOUT=*
// EXEC NOTIFYTS
```

A listing of PROGRAM XOVER

```
//ZMAYABXW JOB (G0109,360,2),AYAU,TIME=(1,30),CLASS=O,MSGCLASS=X
/*JOBPARM LINES=60
// EXEC FORTVC
//SYSIN DD *
C
C FORMAT OF GEOS-3/SEASAT ALTIMETER DATA
C
C VARIABLE    TYPE    DESCRIPTION
C
C     *        I4     SATELITE ID
C              I2     MEASUREMENT TYPE ( 42= OVER LAND , 43 = OVER WATER )
C              I2     TIME SYSTEM ( NM )
C              I4     STATION NUMBER
C              I4     PREPROCESSING INDICATORS
C              I4     MODIFIED JULIAN DATE OF OBSERVATION
C              R8     FRACTION OF DAY PAST MIDNIGHT     (GMT)
C              R8     ALTIMETER OBSERVATION             (METERS)
C     *        I4     SATELLITE GEODETIC LATITUDE       (1E-6 DEGREES)
C     *        I4     SATELLITE EAST LONGITUDE          (1E-6 DEGREES)
C              R4     MEASUREMENT STANDARD DEVIATION    (METERS)
C              I4     NET INSTRUMENT CORRECTION         (MM)
C              I4     METEOROLOGICAL DATA WORD          (GEODYN VOL 3)
C              I4     NET MEDIA CORRECTIONS             (MM)
C              R4     GEOID HEIGHT ABOVE REFERENCE ELLIPSOID (METERS)
C              I4     NET OCEAN DYNAMIC CORRECTIONS     (MM)
C     *        I4     INDICATED SURFACE ELEVATION       (MM)
C     *        I4     S/C REVOLUTION NUMBER
C              I4     MEAN SEA SURFACE ELEVATION (MARSH/MARTIN '81 (MM))
C              I4     DOD REFERENCE RADIAL ORBIT DIFFERENCE (MM)
C              I2     H 1/3                             (CM)
C              I2     AGC                               (DB)
C              I2     WIND SPEED                        (CM/SEC)
C              I2     SURFACE ELEVATION PREPROCESSING WORD
C              I2     DRY TROPOSPHERIC CORRECTION       (MM)
C              I2     FNOC WET TROPOSPHERIC CORRECTION  (MM)
C              I2     SMMR WET TROPOSPHERIC CORRECTION  (MM)
C              I2     IONOSPHERIC CORRECTION            (MM)
C              I2     BAROTROPIC DYNAMIC SEA SURFACE CORRECTION (MM)
C              I2     SOLID EARTH TIDE                  (MM)
C              I2     SCHWIDERSKI OCEAN TIDE            (MM)
C              I2     PARKE OCEAN TIDE                  (MM)
C

      PARAMETER ( SIGMAG = 0.25 )
      PARAMETER ( SIGMAS = 0.10 )
      PARAMETER ( XDMAX  = 100. )
      PARAMETER ( ETOP =   100. )
      PARAMETER ( EBOT =     0. )
      PARAMETER ( SIGE =     2. )
      PARAMETER ( MAXOVR = 2100 )
      PARAMETER ( MPASS  =  142 )
      PARAMETER ( NMAX   =  140 )
      PARAMETER ( MSYM   = (MPASS*(MPASS+1))/2 )
C
      INTEGER*2  I2,I3,I17,I18,I19,I20,I21,I22,I23,I24,I25,I26,I27,I28
      INTEGER*4  I1,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,I15,I16
      REAL*8     R1,R2
      REAL*4     R3,R4
      INTEGER*4  IDSAT
      REAL*4     GLAT, ELON, HSS
      REAL*4     XI(NMAX),YI(NMAX),ZI(NMAX), ZG(NMAX),YR(NMAX)
      REAL*4     SX(NMAX),SY(NMAX),SZ(NMAX), SZG(NMAX),DDZ(NMAX),D(NMAX)
      REAL*4     AX(MAXOVR,MPASS), BX(MAXOVR), XVR(MPASS,MPASS)
      REAL*4     W(MAXOVR), AXT(MPASS,MAXOVR), SCR(MPASS,MPASS)
      REAL*4     AXTBX(MPASS), AXTAX(MSYM)
      REAL*4     X(MPASS,NMAX), H(MPASS,NMAX), HG(MPASS,NMAX)
      REAL*4     SLON1(MPASS), SLAT1(MPASS), SLON2(MPASS), SLAT2(MPASS)
```

```
      REAL*4      U(MPASS), S(MPASS), CIMSL(NMAX-1,3)
      REAL*4      EX(MPASS,MPASS), GY(MPASS,MPASS)
      REAL*8      DAXTBX(MPASS), DAXTAX(MSYM), ERCOVM(MSYM)
      REAL*8      F(NMAX,3),FT(3,NMAX)
      REAL*8      G(NMAX)
      REAL*8      A(MPASS), B(MPASS), C(MPASS)
      INTEGER*4   IDS(MPASS), IDIAG(MPASS), NREV(MPASS), NIREC(MPASS)
      INTEGER*4   ICB(MPASS), ISUM(MPASS), LN(MPASS), LY(MPASS)
      INTEGER*4   ICX(MPASS,MPASS)
      INTEGER     IDP, IREC, IRECA, IXM, NPASS, NREC
      INTEGER     ING, INP, OUP, OUG, OUC
C
      DATA    INP / 3 /, ING / 2 /, OUG / 8 /, OUP / 6 /, OUC / 9 /
      DATA    IREC / 0 /, NREC / 0 /, NPASS / 1 /
C
      G2 = SIGMAG*SIGMAG
      S2 = SIGMAS*SIGMAS
      SMGG = 1.0/(2.*G2)
      SMSS = 1.0/(2.*S2)
      SMSG = 1.0/(S2+G2)
C
      READ ( INP )  I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &              I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &              I20,I21,I22,I23,I24,I25,I26,I27,I28
      IDP = I14
      REWIND INP
C
 1000 READ ( INP, END = 2000 ) I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &                I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &                I20,I21,I22,I23,I24,I25,I26,I27,I28
C
      READ ( ING, 501, END = 2000 )  IDUM1, GL, EL, DUM2, DUM3, DUM4,
     &                               DUM5, DUM6, GUND
C*****
      IF ( I14. EQ. 12793 )  GO TO 999
C*****
      NREC = NREC + 1
      GLAT = 1.0E-6 * FLOAT(I7)
      ELON = 1.0E-6 * FLOAT(I8)
      HSS = 1.0E-3 * FLOAT(I13)
      IDSAT = I1
C
      IF ( I14. EQ. IDP )  THEN
        IF ( HSS. GT. EBOT. AND. HSS. LT. ETOP )  THEN
C
          IREC = IREC + 1
          IDP = I14
          X(NPASS,IREC) = ELON
          H(NPASS,IREC) = HSS
          XI(IREC) = ELON
          YI(IREC) = GLAT
          ZI(IREC) = HSS
          ZG(IREC) = GUND
        END IF
C
      ELSE
C
C.....DETERMINE THE SECOND-ORDER POLYNOMIAL FOR A PASS
C
        IF ( IREC. GE. 5 )    THEN
C
          NREV(NPASS) = IDP
          IDS(NPASS) = IDSAT
          CALL CURVE ( NPASS, XI, YI, ZI, ZG, YR, IREC, IDP,
     &                 IDSAT, A(NPASS), B(NPASS), C(NPASS),
     &                 F, FT, G, SX, SY, SZ, SZG, DDZ, D, SIGE )
          NIREC(NPASS) = IREC
          SLON1(NPASS) = XI(1)
          SLAT1(NPASS) = YI(1)
          SLON2(NPASS) = XI(IREC)
```

```
            SLAT2(NPASS) = YI(IREC)
            DO 100  M = 1, IREC
              X(NPASS,M) = XI(M)
              H(NPASS,M) = ZI(M)
              HG(NPASS,M) = ZG(M)
  100       CONTINUE
C
          ELSE
C
            NPASS = NPASS - 1
C
          END IF
C
          IDP = I14
          IREC = 1
          NPASS = NPASS + 1
          IF ( HSS. GT. EBOT. AND. HSS. LT. ETOP )  THEN
            X(NPASS,IREC) = ELON
            H(NPASS,IREC) = HSS
            XI(IREC) = ELON
            YI(IREC) = GLAT
            ZI(IREC) = HSS
            ZG(IREC) = GUND
          ELSE
            IREC = IREC - 1
          END IF
        END IF
C
  999 CONTINUE
      GO TO 1000
C
 2000 CONTINUE
C
      IF ( IREC. GE. 5 )    THEN
C
        NREV(NPASS) = IDP
        IDS(NPASS) = IDSAT
        CALL CURVE ( NPASS, XI, YI, ZI, ZG, YR, IREC, IDP,
     &               IDSAT, A(NPASS), B(NPASS), C(NPASS),
     &               F, FT, G, SX, SY, SZ, SZG, DDZ, D, SIGE )
        NIREC(NPASS) = IREC
        SLON1(NPASS) = XI(1)
        SLAT1(NPASS) = YI(1)
        SLON2(NPASS) = XI(IREC)
        SLAT2(NPASS) = YI(IREC)
        DO 101  M = 1, IREC
          X(NPASS,M) = XI(M)
          H(NPASS,M) = ZI(M)
          HG(NPASS,M) = ZG(M)
  101   CONTINUE
C
      ELSE
C
      NPASS = NPASS - 1
C
      END IF
C
 3000 WRITE( OUP, * ) ' '
      WRITE( OUP, * ) ' ', NREC, ' RECORDS IN FILE.'
      WRITE( OUP, * ) ' '
      WRITE( OUP, * ) ' ', NPASS, ' PASSES REPRESENTED BY AX**2+BX+C'
C
      CALL XOVER ( NPASS, NREV, IDS, A, B, C, SLON1, SLAT1,
     &             SLON2, SLAT2, ICX, EX, GY, ISUM, LN, LY, IXM )
C
      CALL CLR ( XVR, MPASS*MPASS)
      CALL CLR ( AX, MAXOVR*MPASS)
      CALL CLR ( BX, MAXOVR)
      CALL CLR ( AXTAX, MSYM)
      CALL CLR ( AXTBX, MPASS)
```

```
      CALL DCLR ( DAXTAX, MSYM)
      CALL DCLR ( DAXTBX, MPASS)
C
      WRITE ( OUP, 600 )
      WRITE ( OUP, * ) ' ','   START INTERPOLATION FOR EACH PASS'
C
      DO 200  I = 1, NPASS
        IREC = NIREC(I)
        IF ( X(I,1). LT. X(I,IREC) )  THEN
          DO 201  N = 1, IREC
            XI(N) = X(I,N)
            YI(N) = H(I,N)
 201      CONTINUE
        ELSE
          DO 202  N = 1, IREC
            NM = IREC + 1 - N
            XI(N) = X(I,NM)
            YI(N) = H(I,NM)
 202      CONTINUE
        END IF
        L = 1
        DO 203  N = 2, IREC
          IF ( XI(N). GT. XI(1) )  THEN
            L = L + 1
            XI(L) = XI(N)
            YI(L) = YI(N)
          END IF
 203    CONTINUE
        IRECA = IREC
        IREC = L
        WRITE ( 6,* ) ' '
        WRITE ( 6,* ) ' '
        WRITE ( 6,* ) ' ','SPLINE FITTING TO PASS:',NREV(I),'     ',
     &                    IREC,' DATA POINTS'
        IF ( IRECA. NE. IREC )  THEN
          NJ = IRECA - IREC
          WRITE ( OUP, * ) ' ',NJ,' DATA POINTS THAT ARE NOT IN ',
     &      'ASCENDING OR DECENDING ORDER HAVE BEEN REMOVED.'
        END IF
        CALL ICSCCU ( XI, YI, IREC, CIMSL, IREC-1, IER )
        IF ( IER. NE. 0 )  THEN
          WRITE ( OUP, 609 )  (J,XI(J),YI(J),J=1,IREC)
        END IF
        L = 0
        DO 204  J = 1, NPASS
          IF ( ICX(I,J). NE. 0 )  THEN
            L = L + 1
            U(L) = EX(I,J)
C           WRITE ( OUP, * ) ' ',' PASS:',NREV(J),'    ELON = ',
C    &                        EX(I,J),'   U = ',U(L)
          END IF
 204    CONTINUE
        WRITE ( OUP, * ) ' ',' NUMBER OF CROSSOVERS = ',L
        IF ( L. GT. 0 )  THEN
          CALL ICSEVU ( XI, YI, IREC, CIMSL, IREC-1, U, S, L, IER )
          IF ( IER. NE. 0 )  THEN
            WRITE ( 6,* )  ' ',' ELON ARRAY'
            WRITE ( 6,* )  (XI(MM),MM=1,IREC)
            WRITE ( 6,* )  ' ',' HSS ARRAY'
            WRITE ( 6,* )  (YI(MM),MM=1,IREC)
          END IF
          L = 0
          DO 205  J = 1, NPASS
            IF ( ICX(I,J). NE. 0 )  THEN
              L = L + 1
              XVR(I,J) = S(L)
              WRITE ( OUP, * ) ' ','     XPASS:',NREV(J),'    ELON = ',
     &                           U(L),'  HSS = ',S(L)
            END IF
 205      CONTINUE
```

```
          END IF
  200 CONTINUE
C
          NPASS1 = NPASS - 1
          WRITE ( OUP, 600 )
          WRITE ( OUP, * ) ' ','   SUMMARY OF CROSSOVERS DIFFERENCES ',
         &                 'BEFORE ADJUSTING FOR BIASES:'
          WRITE ( OUP, 608 )  XDMAX
C
          CALL XOVD ( NPASS, IDS, ICX, ISUM, NREV, MAXOVR, XVR, AX, BX,
         &            EX, AXTBX, XDMAX, KC, XRMS,
         &            W, AXT, SMSS, SMGG, SMSG )
C
          DO 300  I = 1, KC
            DO 301  J = IXM+1, NPASS
              AX(I,J-1) = AX(I,J)
  301       CONTINUE
  300     CONTINUE
C
          DO 400  I = 1, KC
            DO 401  J = 1, NPASS1
              AXT(J,I) = AX(I,J) * W(I)
  401       CONTINUE
  400     CONTINUE
C
          CALL VMULFF ( AXT, AX, NPASS1, KC, NPASS1, MPASS, MAXOVR, SCR,
         &              MPASS, IER )
          CALL VCVTFS ( SCR, NPASS1, MPASS, AXTAX )
C
          CALL VMULFF ( AXT, BX, NPASS1, KC, 1, MPASS, MAXOVR, AXTBX,
         &              MPASS, IER )
C
          DO 700  I = 1, NPASS1
            IDIAG(I) = ( I * (I+1) ) / 2
  700     CONTINUE
          DO 701  I = 1, IDIAG(NPASS1)
            DAXTAX(I) = DBLE(AXTAX(I))
  701     CONTINUE
          DO 702  I = 1, NPASS1
            DAXTBX(I) = DBLE(AXTBX(I))
  702     CONTINUE
          CALL SOLVE ( DAXTAX, DAXTBX, IDIAG, NPASS1, .TRUE., .TRUE. )
C
          DO 800  I = IXM, NPASS1
            K = NPASS + IXM - I
            DAXTBX(K) = DAXTBX(K-1)
  800     CONTINUE
          DAXTBX(IXM) = 0.
C
          DO 801  I = 1, NPASS
            AXTBX(I) = SNGL(DAXTBX(I))
  801     CONTINUE
C
          XRB = 0.
          DO 802  I = 1, NIREC(IXM)
            XRB = XRB + H(IXM,I) - HG(IXM,I)
  802     CONTINUE
          XRB = XRB / FLOAT(NIREC(IXM))
C
          DO 803  I = 1, NPASS
            AXTBX(I) = AXTBX(I) + XRB
  803     CONTINUE
C
          WRITE ( OUP, 600 )
          WRITE ( OUP, * ) ' ','   SUMMARY OF CROSSOVERS DIFFERENCES ',
         &                 'AFTER ADJUSTING FOR BIASES:'
          WRITE ( OUP, 608 )  XDMAX
C
          CALL XOVD ( NPASS, IDS, ICX, ISUM, NREV, MAXOVR, XVR, AX, BX,
         &            EX, AXTBX, 1000.0, KC, XRMS,
```

```
     &             W, AXT, SMSS, SMGG, SMSG )
C
      WRITE ( OUP, 601 )  NREV(IXM), XRB
      DO 900  I = 1, NPASS
        IF ( ISUM(I). EQ. 0 )  THEN
          AXTBX(I) = 0.
          WRITE ( OUP, 603 )  I, NREV(I), AXTBX(I)
C       ELSE IF ( I. EQ. IXM )  THEN
C         WRITE ( OUP, 604 )  IXM, NREV(IXM)
        ELSE
          WRITE ( OUP, 602 )  I, NREV(I), AXTBX(I)
        END IF
  900 CONTINUE
C
      DO 1100  I = 1, NPASS
        ICB(I) = IFIX(AXTBX(I)*1000.)
 1100 CONTINUE
C
      REWIND INP
      NREC = 0
 4000 READ ( INP, END = 9000 ) I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &                 I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &                 I20,I21,I22,I23,I24,I25,I26,I27,I28
C*****
      IF ( I14. EQ. 12793 )  THEN
        I13 = -1000000
        GO TO 888
      END IF
C*****
      NREC = NREC + 1
      IBIAS = 0
C
      DO 1101  I = 1, NPASS
        IF ( I14. EQ. NREV(I) )  THEN
          IBIAS = I
          GO TO 5000
        END IF
 1101 CONTINUE
C
 5000 CONTINUE
C
      IF ( IBIAS. EQ. 0 )  THEN
        I13 = -1000000
      ELSE
        I13 = I13 - ICB(IBIAS)
      END IF
C
  888 CONTINUE
      WRITE ( OUG ) I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &              I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &              I20,I21,I22,I23,I24,I25,I26,I27,I28
      GO TO 4000
C
 9000 WRITE ( OUP, 600 )
      WRITE ( OUP, * ) ' ', NREC, ' RECORDS ADJUSTED FOR ORBIT BIAS'
C
C.....DETERMINE THE ERROR COVARIANCE MATRIX
C
      CALL SYMINV ( DAXTAX, DAXTBX, IDIAG, NPASS1, ERCOVM )
C
      MM = (NPASS1*(NPASS1+1))/2
      WRITE ( OUC, 610 )  (ERCOVM(I),I=1,MM)
C
      WRITE ( OUP, * ) ' '
      WRITE ( OUP, * ) ' ','THE COVARIANCE MATRIX OF BIAS ADJUSTMENT',
     &                     ' HAS BEEN WRITTEN TO FILE BLACK.PCOV.'
C
  501 FORMAT(2X,I3,8F9.3)
  600 FORMAT(1H1)
  601 FORMAT(1H1,T20,'BIAS FOR SATELLITE PASSES OVER THE ',
```

```
     &            'BLACK SEA:',//,T23,'THE REFERENCE SATELLITE ',
     &            'PASS IS # ',I5,//,T26,'THE BIAS OF THIS PASS WRT ',
     &            'GEOID MODEL OSU-180 IS ',F10.4,' METERS',//)
  602 FORMAT(1H0,T15,I4,5X,'PASS #',I6,5X,'BIAS (METERS):',F10.4)
  603 FORMAT(1H0,T15,I4,5X,'PASS #',I6,5X,'BIAS (METERS):',F10.4,5X,
     &       '**** THIS PASS HAS NO CROSSOVER ****')
  604 FORMAT(1H0,T15,I4,5X,'PASS #',I6,5X,' ### REFERENCE PASS ###')
  605 FORMAT((1H ,32F4.0))
  606 FORMAT((1H ,26F5.0))
  607 FORMAT((1H ,10(1PE12.4)))
  608 FORMAT(1H ,/,T10,'THE CUTOFF WINDOW FOR CROSSOVER DIFFERENCES',
     &                 ' IS ',F7.2,' METERS',/)
  609 FORMAT((1H ,T10,I6,2F12.4))
  610 FORMAT((4D20.13))
C
      STOP
      END
C
      SUBROUTINE XOVD ( NPASS, IDS, ICX, ISUM, NREV, MAXOVR, XVR,
     &                  AX, BX, EX, BIAS, XDMAX, KC, XRMS,
     &                  W, AXT, SMSS, SMGG, SMSG )
C
      REAL*4     XVR(NPASS,NPASS), EX(NPASS,NPASS)
      REAL*4     BIAS(NPASS), AX(MAXOVR,NPASS), BX(MAXOVR)
      REAL*4     W(MAXOVR), AXT(NPASS,MAXOVR)
      INTEGER*4 ICX(NPASS,NPASS), IDS(NPASS), ISUM(NPASS), NREV(NPASS)
      INTEGER*4 K75, K78, K7578, KC, MAXOVR, OUP
      DATA      OUP / 6 /
C
      XRMS = 0.
      KC = 0
      K75 = 0
      K78 = 0
      K7578 = 0
C
      DO 100  I = 1, NPASS
        DO 101  J = I, NPASS
          IF ( ICX(I,J). NE. 0 )  THEN
            XVR(I,J) = XVR(I,J) - BIAS(I)
            XVR(J,I) = XVR(J,I) - BIAS(J)
            BXT = XVR(I,J) - XVR(J,I)
            IF ( ICX(I,J). EQ. -1 )  THEN
              ICX(I,J) = 0
              ICX(J,I) = 0
              WRITE ( OUP, 600 )  EX(I,J),NREV(I),XVR(I,J),NREV(J),
     &                            XVR(J,I),BXT
            ELSE IF ( ABS(BXT). LE. XDMAX )  THEN
              KC = KC + 1
              AX(KC,I) = 1.0
              AX(KC,J) = -1.0
              BX(KC) = BXT
              XRMS = XRMS + BXT*BXT
              WRITE ( OUP, 601 )  EX(I,J),NREV(I),XVR(I,J),NREV(J),
     &                            XVR(J,I),BX(KC)
              IF ( IDS(I). EQ. IDS(J) )  THEN
                IF ( IDS(I). EQ. 7502701 )  THEN
                  K75 = K75 + 1
                  W(KC) = SMGG
                END IF
                IF ( IDS(I). EQ. 7806401 )  THEN
                  K78 = K78 + 1
                  W(KC) = SMSS
                END IF
              ELSE
                K7578 = K7578 + 1
                W(KC) = SMSG
              END IF
            ELSE
              ICX(I,J) = 0
              ICX(J,I) = 0
```

```
                ISUM(I) = ISUM(I) - 1
                ISUM(J) = ISUM(J) - 1
                WRITE ( OUP, 602 )  EX(I,J),NREV(I),XVR(I,J),NREV(J),
     &                              XVR(J,I),BXT
              END IF
            END IF
  101     CONTINUE
  100   CONTINUE
C
        XRMS =  SQRT ( XRMS / FLOAT(KC) )
C
        WRITE ( OUP, 603 )  XRMS, K75, K78, K7578, KC
C
  600   FORMAT(1H ,T15,'X @ E. LONG = ',F6.2,8X,
     &           '#',I5,'(',F6.2,')'   -   #',I5,'(',F6.2,')',
     &           '      ----->  ',F8.2,5X,' ** EDITTED OUT (A PRIORI) **')
  601   FORMAT(1H ,T15,'X @ E. LONG = ',F6.2,8X,
     &           '#',I5,'(',F6.2,')'   -   #',I5,'(',F6.2,')',
     &           '      ----->  ',F8.2)
  602   FORMAT(1H ,T15,'X @ E. LONG = ',F6.2,8X,
     &           '#',I5,'(',F6.2,')'   -   #',I5,'(',F6.2,')',
     &           '      ----->  ',F8.2,5X,' ** DISCARDED (IN SITU) **')
  603   FORMAT(1H0,/,T20,'RMS (M) = ',F10.4,//,
     &             T24,'   # OF  GEO3-GEO3    CROSSOVERS = ',I5,//,
     &             T24,'   # OF SEASAT-SEASAT CROSSOVERS = ',I5,//,
     &             T24,'   # OF  GEO3-SEASAT  CROSSOVERS = ',I5,//,
     &             T24,'TOTAL # OF CONSIDERED CROSSOVERS = ',I5)
C
        RETURN
        END
C
        SUBROUTINE CURVE ( NPASS, X, Y, Z, ZG, YR, IREC, IDP, IDSAT,
     &                     A, B, C, F, FT, G,
     &                     SX, SY, SZ, SZG, DDZ, D, SIGE )
C
        REAL*4        X(IREC), Y(IREC), Z(IREC), ZG(IREC), YR(IREC)
        REAL*4        SX(IREC), SY(IREC), SZ(IREC)
        REAL*4        SZG(IREC), DDZ(1REC), D(IREC)
        REAL*8        F(IREC,3), FT(3,IREC)
        REAL*8        E(3,3), EI(3,3), AE(6)
        REAL*8        G(IREC), H(3)
        REAL*8        A, B, C, DET, YD
        INTEGER*4     IDP, IDSAT, IREC, NPASS
        INTEGER*4     IDIAG(3) / 1, 3, 6 /
        INTEGER*4     OUP / 6 /
C
C.....EVICT THE BAD DATA POINTS BY COMPARING THE ABSOLUTE DIFFERENCE
C       BETWEEN ADJACENT DATA POINTS
C
        NB = 0
        IREC1 = IREC - 1
C
        DO 111  I = 1, IREC
          SX(I) = -1.0
  111   CONTINUE
C
        DO 112  I = 1, IREC1
          DDZ(I) = ABS( Z(I+1) - Z(I) )
  112   CONTINUE
C
        DO 113  J = 1, IREC1
          IF(DDZ(J).LT.SIGE)  THEN
            K = J
            GO TO 200
          END IF
  113   CONTINUE
C
  200   CONTINUE
        DO 222  J = 1, IREC
          IF(DDZ(IREC-J).LT.SIGE)  THEN
```

```
            IRECG = IREC-J
            GO TO 300
          END IF
  222 CONTINUE
C
  300 KB = 0
      DO 333  I = 1, IRECG-1
        IF(DDZ(I).GT.SIGE)  KB = KB + 1
  333 CONTINUE
C
      LG = 1
      SX(LG) = X(K)
      SY(LG) = Y(K)
      SZ(LG) = Z(K)
      SZG(LG) = ZG(K)
C
  400 LG = LG + 1
      K = K + 1
      IF(DDZ(K-1).GT.SIGE)  THEN
        K = K + 1
        DO 444  I = 1, KB-1
          IF(DDZ(K-1).GT.SIGE)  K = K + 1
  444   CONTINUE
      END IF
      SX(LG) = X(K)
      SY(LG) = Y(K)
      SZ(LG) = Z(K)
      SZG(LG) = ZG(K)
      IF(K.LE.IRECG)  GO TO 400
C
      DO 555  I = 1, LG
        F(I,1) = SX(I)*SX(I)
        F(I,2) = SX(I)
        F(I,3) = 1.0D0
        G(I) = SY(I)
  555 CONTINUE
C
      DO 666  I = 1, LG
        DO 667  J = 1, 3
          FT(J,I) = F(I,J)
  667   CONTINUE
  666 CONTINUE
C
      DO 777  I = 1, 3
        DO 778  J = 1, 3
          E(I,J) = 0.D0
          DO 779  K = 1, LG
            E(I,J) = E(I,J) + FT(I,K)*F(K,J)
  779     CONTINUE
  778   CONTINUE
  777 CONTINUE
C
      DO 888  I = 1, 3
        H(I) = 0.D0
        DO 889  K = 1, LG
          H(I) = H(I) + FT(I,K)*G(K)
  889   CONTINUE
  888 CONTINUE
C
      AE(1) = E(1,1)
      AE(2) = E(1,2)
      AE(3) = E(2,2)
      AE(4) = E(1,3)
      AE(5) = E(2,3)
      AE(6) = E(3,3)
C
      CALL SOLVE ( AE, H, IDIAG, 3, .TRUE., .TRUE. )
C
      A = H(1)
      B = H(2)
```

```
      C = H(3)
C
C     DET = E(1,1) * ( E(2,2)*E(3,3) - E(2,3)*E(2,3) ) +
C    &      E(1,2) * ( E(1,3)*E(2,3) - E(1,2)*E(3,3) ) +
C    &      E(1,3) * ( E(1,2)*E(2,3) - E(1,3)*E(2,2) )
C
C     EI(1,1) = ( E(2,2)*E(3,3) - E(2,3)*E(2,3) ) / DET
C     EI(2,2) = ( E(1,1)*E(3,3) - E(1,3)*E(1,3) ) / DET
C     EI(3,3) = ( E(1,1)*E(2,2) - E(1,2)*E(1,2) ) / DET
C     EI(2,3) = ( E(1,2)*E(1,3) - E(1,1)*E(2,3) ) / DET
C     EI(1,3) = ( E(1,2)*E(2,3) - E(1,3)*E(2,2) ) / DET
C     EI(1,2) = ( E(1,3)*E(2,3) - E(1,2)*E(3,3) ) / DET
C
C     A = ( EI(1,1)*H(1) + EI(1,2)*H(2) + EI(1,3)*H(3) )
C     B = ( EI(1,2)*H(1) + EI(2,2)*H(2) + EI(2,3)*H(3) )
C     C = ( EI(1,3)*H(1) + EI(2,3)*H(2) + EI(3,3)*H(3) )
C
      RMS = 0.
      DO 999  I = 1, LG
        YD = A*SX(I)*SX(I) + B*SX(I) + C
        YR(I) = SNGL(YD)
        D(I) = YR(I) - SY(I)
        RMS = RMS + D(I)*D(I)
  999 CONTINUE
      RMS = SQRT ( RMS/FLOAT(LG) )
C
      WRITE ( OUP, 601 )  NPASS, IDP, IDSAT, A, B, C,
     &                    SX(1), SY(1), SX(LG), SY(LG), RMS, LG
      IF ( RMS. GT. 0.005 )  THEN
        WRITE ( 6, * )  ' '
        WRITE ( 6, 602 )  (M,SX(M),SY(M),YR(M),D(M),M=1,LG)
        WRITE ( 6, * )  ' '
      END IF
C
      IF ( LG. NE. IREC )  THEN
        IREC = LG
        DO 1111  I = 1, IREC
          X(I) = SX(I)
          Y(I) = SY(I)
          Z(I) = SZ(I)
          ZG(I) = SZG(I)
 1111   CONTINUE
      END IF
C
  601 FORMAT(1H0,//,1X,I3,I7,I9,3E14.5,5F10.4,I6)
  602 FORMAT((1H ,T5,I4,3X,3F10.4,2X,F10.4))
C
      RETURN
      END
C
      SUBROUTINE XOVER ( NPASS, NREV, IDSAT, A, B, C, SLON1, SLAT1,
     &                   SLON2, SLAT2, ICX, EX, GY, ISUM, LN, LY, IXM )
C
      INTEGER     NREV(NPASS), IDSAT(NPASS)
      INTEGER     ICX(NPASS,NPASS), ISUM(NPASS), LN(NPASS), LY(NPASS)
      INTEGER     INDEX(2)
      REAL*4      SLAT1(NPASS), SLAT2(NPASS), SLON1(NPASS), SLON2(NPASS)
      REAL*4      EX(NPASS,NPASS), GY(NPASS,NPASS)
      REAL*4      P, Q
      REAL*8      A(NPASS), B(NPASS), C(NPASS)
      REAL*8      T(2)
      REAL*8      DA,DB,DC,RT,SRT
      INTEGER     ITOTAL, IXM, MAXIX, MX, MXS, MX75, MX78, NPASS
      INTEGER     IJK, IP, JP
      INTEGER     JOUT, OUP
      DATA        JOUT / 4 /, OUP / 6 /
      DATA        MAXIX / 0 /, MX / 0 /, MX75 / 0 /, MX78 / 0 /
      DATA        ITOTAL / 0 /
C
      DO 123  I = 1, NPASS
```

```
          ISUM(I) = 0
          DO 234  J = 1, NPASS
            ICX(I,J) = 0
            EX(I,J) = 0.
            GY(I,J) = 0.
  234     CONTINUE
  123   CONTINUE
C
        WRITE ( OUP, 600 )
C
        DO 222  I = 1, NPASS-1
C
          WRITE( OUP, 601 )  NREV(I), IDSAT(I), SLON1(I), SLAT1(I),
     &                       SLON2(I), SLAT2(I)
C
          DO 333  J = I+1, NPASS
C
            INDEX(1) = 0
            INDEX(2) = 0
            DB = B(J) - B(I)
            DA =  A(I) - A(J)
            DC =  C(I) - C(J)
            RT = DB*DB - 4.0*DA*DC
C
            IF ( RT. GE. 0.0 )  THEN
              IF ( RT. EQ. 0.0 )  THEN
                SRT = 0.0
              ELSE
                SRT = DSQRT(RT)
              END IF
              T(1) = (DB+SRT)/(2.0*DA)
              T(2) = (DB-SRT)/(2.0*DA)
C             WRITE(6,*)  T, RT, SRT
C
              DO 135  L = 1, 2
                IF ( SLON2(I). GT. SLON1(I) ) THEN
                  IF( T(L). GE. SLON1(I). AND. T(L). LE. SLON2(I) )
     &              INDEX(L) = INDEX(L) + 1
                ELSE
                  IF( T(L). LE. SLON1(I). AND. T(L). GE. SLON2(I) )
     &              INDEX(L) = INDEX(L) + 1
                END IF
                IF ( SLON2(J). GT. SLON1(J) ) THEN
                  IF( T(L). GE. SLON1(J). AND. T(L). LE. SLON2(J) )
     &              INDEX(L) = INDEX(L) + 1
                ELSE
                  IF( T(L). LE. SLON1(J). AND. T(L). GE. SLON2(J) )
     &              INDEX(L) = INDEX(L) + 1
                END IF
  135         CONTINUE
C
              DO 246  L = 1, 2
                IF ( INDEX(L). EQ. 2 )  THEN
                  MX = MX + 1
                  P = SNGL (T(L) )
                  Q = SNGL ( A(I)*P*P + B(I)*P + C(I) )
                  EX(I,J) = P
                  GY(I,J) = Q
                  EX(J,I) = P
                  GY(J,I) = Q
                  WRITE ( OUP, 602)  NREV(J), IDSAT(J), P, Q,
     &                               SLON1(J),SLAT1(J),SLON2(J),SLAT2(J)
                  ICX(I,J) = ICX(I,J) + 1
                  ICX(J,I) = ICX(I,J)
                  IF ( IDSAT(I). EQ. IDSAT(J) )  THEN
                    IF ( IDSAT(I). EQ. 7502701 )  MX75 = MX75 + 1
                    IF ( IDSAT(I). EQ. 7806401 )  MX78 = MX78 + 1
                  END IF
                END IF
  246         CONTINUE
```

```
C
          END IF
C
  333   CONTINUE
C
  222 CONTINUE
C
      WRITE ( OUP, * )  '  '
      WRITE ( OUP, * )  '  '
      WRITE ( OUP, * )  '  TOTAL # OF POSSIBLE CROSSOVERS = ', MX
      WRITE ( OUP, * )  '  '
      WRITE ( OUP, * )  '  TOTAL # OF GEO3-GEO3 CROSSOVERS = ', MX75
      WRITE ( OUP, * )  '  '
      WRITE ( OUP, * )  '  TOTAL # OF SEASAT-SEASAT CROSSOVERS = ', MX78
      MXS = MX - MX75 - MX78
      WRITE ( OUP, * )  '  '
      WRITE ( OUP, * )  '  TOTAL # OF GEOS3-SEASAT CROSSOVERS = ', MXS
C
C.....EDIT OUT BAD CROSSOVERS BASED ON A PRIORI KNOWLEDGE
C
      DO 555  IR = 1, 500
        READ ( JOUT, 501, END = 556 )  JC, IP, JP
        IF ( JC. EQ. 0. AND. ICX(IP,JP). NE. 0 )  THEN
          ICX(IP,JP) = -1
          ICX(JP,IP) = -1
        END IF
  555 CONTINUE
C
  556 CONTINUE
C
      DO 345  I = 1, NPASS
        DO 456  J = 1, NPASS
          IJK = ICX(I,J)
          IF ( IJK. LT. 0 )  IJK = 0
          ISUM(I) = ISUM(I) + IJK
  456   CONTINUE
        ITOTAL = ITOTAL + ISUM(I)
  345 CONTINUE
      ITOTAL = ITOTAL/2
C
      DO 567  I = 1, NPASS
        MAXIX = MAX0 ( MAXIX, ISUM(I) )
  567 CONTINUE
C
      DO 678  I = 1, NPASS
        IF ( ISUM(I). EQ. MAXIX )  IXM = I
  678 CONTINUE
C
      WRITE ( OUP, 603 )
      WRITE ( OUP, 604 )  ( I, NREV(I), I = 1, NPASS )
      WRITE ( OUP, 605 )  NPASS
      WRITE ( OUP, 606 )  ( NREV(I),(ICX(I,J),J=1,94),ISUM(I),
     &                      I=1,NPASS )
      WRITE ( OUP, 607 )  NREV(IXM), MAXIX, ITOTAL
C
      MM = 0
      NN = 0
      DO 789  I = 1, NPASS
        IF ( ICX(IXM,I). EQ. 1 )  THEN
          MM = MM + 1
          LY(MM) = I
        ELSE
          NN = NN + 1
          LN(NN) = I
        END IF
  789 CONTINUE
C
      WRITE ( OUP, 608 )  NREV(IXM), MM, NN, MM+NN
      WRITE ( OUP, 609 )  ( NREV(LY(I)), I = 1, MM )
C
```

```
      DO 987  J = 1, NN
        WRITE ( OUP, 610 )  NREV(LN(J))
        DO 876  I = 1, NPASS
          IF ( ICX(I,LN(J)). EQ. 1 )  THEN
            WRITE ( OUP, 611 )  NREV(I)
            GO TO 888
          END IF
  876     CONTINUE
  888   CONTINUE
  987 CONTINUE
C
  501 FORMAT(I1,2I10)
  600 FORMAT(1H1,T20,'BLACK SEA CROSS-OVER LISTING:',//)
  601 FORMAT(1H0,//,T5,'PASS = ',I6,5X,'SAT ID = ',I10,5X,4F10.6//)
  602 FORMAT(1H ,T10,'XPASS = ',I6,5X,'SAT ID = ',I10,5X,2F10.3,
     &       5X,4F10.6)
  603 FORMAT(1H1,T20,'ROW-COLUMN ELEMENTS OF PASS CORRELATION MATRIX:',
     &       //)
  604 FORMAT((1H ,10(3X,I4,I6)))
  605 FORMAT(1H1,T20,'PARTIAL (94) PASS CORRELATION MATRIX FOR',
     &           I5,' PASSES:',//)
  606 FORMAT((1H ,10X,I8,5X,94I1,5X,I3))
  607 FORMAT(1H0,//,T10,'PASS # ',I6,' HAS THE MAXIUM OF ',I5,
     &       ' CROSSOVERS.',//,T10,'THE TOTAL # OF CROSSOVERS IS:',I5)
  608 FORMAT(1H1,T5,'REFERENCE PASS = ',I6,T110,3I6,/
     &           T10,'THE DIRECT CROSS-PASSES ARE:',//)
  609 FORMAT((1H0,20I6))
  610 FORMAT(1H0,//,T5,'PASS: ',I6,'  IS INDIRECTLY RELATED TO THE')
  611 FORMAT(1H ,T10,'REFERENCE PASS VIA PASS ',I6)
C
      RETURN
      END
C
      SUBROUTINE SOLVE (A,B, IDIAG,NEQ, FACT,BACK)
C
C  Compute the U**T * D * U  factorization of the symmetric matrix
C  stored in A, if FACT = TRUE;  and solve A * X = B if BACK = TRUE.
C
C
C  A          Contains the compacted-column form of the upper triangular
C             part of the coefficient matrix.  After factorization, it
C             contains D and U.
C  B          Right-hand-side vector.  After backsubstitution, it
C             contains the solution.
C
C  IDIAG      Addresses of the diagonal terms in A.
C  NEQ        Number of equations
C
C  FACT       If FACT = TRUE, then factor A;  otherwise do not factor A.
C  BACK       If BACK = TRUE, reduce B and backsubstitute;  otherwise
C             do not solve the equations.
C
C
C
      IMPLICIT REAL*8    (A-H,O-Z)
      LOGICAL            FACT,BACK
      DIMENSION          A(1),B(1),IDIAG(1)
C
C
C  Factor A, reduce B
      JR = 0
      DO 400  J = 1, NEQ
        JD = IDIAG(J)
        JH = JD - JR
        IS = J - JH + 2
C
        IF (JH .LT. 2)  GOTO 390
C
        IF (FACT) THEN
C
```

```
            IF (JH .GT. 2) THEN
C..         Reduce column J rows IS to J-1: do not divide by row diagonal
              K = JR + 2
              ID = IDIAG(IS - 1)
C
              DO 100 I = IS, J-1
                IR = ID
                ID = IDIAG(I)
                IH = MIN (ID-IR-1,I-IS+1)
                IF (IH.GT.0)  A(K) = A(K) - DOT(A(K-IH),A(ID-IH),IH)
                K = K + 1
  100         CONTINUE
            ENDIF
C
C..         Divide by row diagonal, and reduce diagonal term in column J
C
            IR = JR + 1
            K = J - JD
            DO 200  I = IR, JD-1
              ID = IDIAG(K+I)
              IF (A(ID).EQ.0.0) GOTO 200
              D = -A(I)
              A(I) = A(I)/A(ID)
              A(JD) = A(JD) + D*A(I)
  200       CONTINUE
C
          ENDIF
C
C..       Reduce RHS
          IF (BACK)  B(J) = B(J) - DOT(A(JR+1),B(IS-1),JH-1)
C
  390     JR = JD
  400   CONTINUE
C
C
        IF (.NOT.BACK)  RETURN
C
C
C   Divide by diagonal pivots
C
        DO 700 I = 1,NEQ
          ID = IDIAG(I)
          IF (A(ID).NE.0.0)    B(I) = B(I)/A(ID)
  700   CONTINUE
C
C   Backsubstitute
C
        J = NEQ
        JD = IDIAG(J)
C
  801   D = -B(J)
        J = J - 1
        IF (J.LE.0)  RETURN
C
        JR = IDIAG(J)
        IF (JD-JR.GT.1) THEN
          IS = J - JD + JR + 2
          K = JR - IS + 1
          DO 810  I = IS, J
  810     B(I) = B(I) + A(I+K)*D
        ENDIF
C
        JD = JR
        GOTO 801
C
        END
C
        FUNCTION DOT (A,B,N)
C
C   Compute the dot product of the two N-vectors A and B.
```

```
C
      INTEGER*4  N, I
      REAL*8     DOT, A(1),B(1)
C
      DOT = 0.0
C
      DO 100  I = 1, N
        DOT = DOT + A(I)*B(I)
  100 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE CLR (A,NA)
C
      INTEGER*4  NA, I
      REAL*4     A(1)
C
      DO 100  I = 1, NA
        A(I)=0.0
  100 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE DCLR (A,NA)
C
      INTEGER*4  NA, I
      REAL*8     A(1)
C
      DO 100  I = 1, NA
        A(I)=0.0D0
  100 CONTINUE
C
      RETURN
      END
      SUBROUTINE SYMINV ( A, B, IDIAG, NEQ, AINV )
C
      IMPLICIT REAL*8   (A-H,O-Z)
      DIMENSION   A(1), B(1), AINV(1), IDIAG(1)
C
      K = 0
      DO 100  I = 1, NEQ
        CALL DCLR ( B, NEQ )
        B(I) = 1.0
        CALL SOLVE ( A, B, IDIAG, NEQ, .FALSE., .TRUE. )
        DO 110  J = 1, I
          K = K + 1
          AINV(K) = B(J)
  110   CONTINUE
  100 CONTINUE
C
      RETURN
      END
//*
// EXEC LINKGOV,REGION=5000K
//SYSLIB     DD DSN=SYS2.IMSLS,DISP=SHR
//*          DD DSN=SYS2.IMSLD,DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** *
//*                                                        *
//*                    BLACK SEA DATA                      *
//*                                                        *
//* **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT02F001 DD DSN=ZMAYA.ALTIM.DATA(BLKRAPP),DISP=SHR
//GO.FT03F001 DD DSN=ZMAYA.BLACK.DATA,DISP=SHR
//GO.FT04F001 DD DSN=ZMAYA.BLKPOUT.DATA,DISP=SHR
//GO.FT06F001 DD SYSOUT=*
//GO.FT08F001 DD DSN=ZMAYA.BLACKA.DATA,DISP=SHR
```

```
//GO.FT09F001 DD DSN=ZMAYA.BLACK.PCOV,DISP=SHR
// EXEC NOTIFYTS
```

A listing of PROGRAM PLTAEP

```
//ZMAYABEA JOB (G0109,360,75),AYAU,TIME=(12,00),CLASS=O,MSGCLASS=X
/*JOBPARM LINES=60
//TEMPLATE EXEC G38PLOT
//FORT.SYSIN DD *
C
C  FORMAT OF GEOS-3/SEASAT ALTIMETER DATA
C  VARIABLE    TYPE     DESCRIPTION
C                I4     SATELITE ID
C                I2     MEASUREMENT TYPE ( 42= OVER LAND , 43 = OVER WATER )
C                I2     TIME SYSTEM ( NM )
C                I4     STATION NUMBER
C                I4     PREPROCESSING INDICATORS
C      *         I4     MODIFIED JULIAN DATE OF OBSERVATION
C      *         R8     FRACTION OF DAY PAST MIDNIGHT     (GMT)
C                R8     ALTIMETER OBSERVATION             (METERS)
C                I4     SATELLITE GEODETIC LATITUDE       (1E-6 DEGREES)
C                I4     SATELLITE EAST LONGITUDE          (1E-6 DEGREES)
C                R4     MEASUREMENT STANDARD DEVIATION    (METERS)
C                I4     NET INSTRUMENT CORRECTION         (MM)
C                I4     METEOROLOGICAL DATA WORD          (GEODYN VOL 3)
C                I4     NET MEDIA CORRECTIONS             (MM)
C                R4     GEOID HEIGHT ABOVE REFERENCE ELLIPSOID (METERS)
C                I4     NET OCEAN DYNAMIC CORRECTIONS     (MM)
C      *         I4     INDICATED SURFACE ELEVATION       (MM)
C      *         I4     S/C REVOLUTION NUMBER
C                I4     MEAN SEA SURFACE ELEVATION (MARSH/MARTIN '81 (MM))
C                I4     DOD REFERENCE RADIAL ORBIT DIFFERENCE (MM)
C                I2     H 1/3                             (CM)
C                I2     AGC                               (DB)
C                I2     WIND SPEED                        (CM/SEC)
C                I2     SURFACE ELEVATION PREPROCESSING WORD
C                I2     DRY TROPOSPHERIC CORRECTION       (MM)
C                I2     FNOC WET TROPOSPHERIC CORRECTION  (MM)
C                I2     SMMR WET TROPOSPHERIC CORRECTION  (MM)
C                I2     IONOSPHERIC CORRECTION            (MM)
C                I2     BAROTROPIC DYNAMIC SEA SURFACE CORRECTION (MM)
C                I2     SOLID EARTH TIDE                  (MM)
C                I2     SCHWIDERSKI OCEAN TIDE            (MM)
C                I2     PARKE OCEAN TIDE                  (MM)
C
      PARAMETER ( MAXPLT = 9999 )
      PARAMETER ( NMAX = 2500 )
      PARAMETER ( NINT =  150 )
      PARAMETER ( ETOP =  100. )
      PARAMETER ( EBOT =    0. )
      PARAMETER ( SIGE = 2.0 )
      PARAMETER ( SIGT = 8.0E-5 )
      PARAMETER ( IC = NMAX - 1 )
C
      INTEGER*2  I2,I3,I17,I18,I19,I20,I21,I22,I23,I24,I25,I26,I27,I28
      INTEGER*4  I1,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,I15,I16
      REAL*8     R1,R2
      REAL*4     R3,R4
      INTEGER*4  IDSAT
      REAL*4     GLAT, ELON, HSS
      REAL*4     X(NMAX),Y(NMAX),YM(NMAX),DF(NMAX),C(NMAX-1,3)
      REAL*4     F(NMAX), GX(NMAX), GY(NMAX), DGY(NMAX), DDY(NMAX)
      REAL*4     DDT(NMAX), XA(NMAX), YA(NMAX), DYA(NMAX)
      REAL*4     WK(7*NMAX+14), XF(NINT+1), YF(NINT+1)
      INTEGER    INP, IREC, IPLOT, NREC, NREV, OUP
      INTEGER    LIST(6)
      INTEGER    IBC(NMAX)
C
      DATA   INP / 3 /, ING / 4 /, OUP / 6 /
      DATA   LIST / 8, 110, 75, 0, 7, 10 /
      DATA   IREC / 0 /, NREC / 0 /, IPLOT / 0 /
C
```

```
      READ ( INP )  I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &              I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &              I20,I21,I22,I23,I24,I25,I26,I27,I28
      NREV = I14
      REWIND INP
      CALL USTART
      CALL UINQES(1.0,SUPRT)
      IF(SUPRT.NE.0.0)  CALL UESCAP(1.0,LIST,6.0,IDUM,DUM)
C
 1000 READ ( INP, END = 2000 ) I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     *                I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     *                I20,I21,I22,I23,I24,I25,I26,I27,I28
      READ ( ING, 500, END = 2000 )  IDUM1,GL,EL,DUM2,DUM3,DUM4,
     *                               DUM5,DUM6,GUND
  500 FORMAT(2X,I3,8F9.3)
      NREC = NREC + 1
      GLAT = 1.0E-6 * FLOAT(I7)
      ELON = 1.0E-6 * FLOAT(I8)
      HSS  = 1.0E-3 * FLOAT(I13)
      IDSAT = I1
C
      IF ( I14. EQ. NREV )  THEN
        IF ( HSS. GT. EBOT. AND. HSS. LT. ETOP )  THEN
          IREC = IREC + 1
          RMJD = FLOAT(I6)
          RNREV = FLOAT(I14)
          X(IREC) = R1
          Y(IREC) = HSS
          DF(IREC) = R3
          YM(IREC) = GUND
        END IF
      ELSE
C
C.....PLOT THE PREVIOUS ELEVATION PROFILE
C
        IF ( IPLOT. GT. MAXPLT )  GO TO 3000
        IF ( IREC. GT. 4 )  THEN
C
          IPLOT = IPLOT + 1
          WRITE ( OUP, 601 )  IPLOT, RNREV
          CALL PLOT(X,Y,YM,DF,IREC,RMJD,RNREV,IDSAT,IC,NINT,NMAX,
     &              C,F,GX,GY,DGY,DDY,WK,XF,YF,DDT,XA,YA,DYA,IBC,
     &              SIGE,SIGT)
C
        END IF
C
        NREV = I14
        IREC = 1
        IF ( HSS. GT. EBOT. AND. HSS. LT. ETOP )  THEN
          X(IREC) = R1
          Y(IREC) = HSS
          DF(IREC) = R3
          YM(IREC) = GUND
        ELSE
          IREC = IREC - 1
        END IF
      END IF
C
      GO TO 1000
C
 2000 CONTINUE
C
      IF ( IREC. GT. 4 )  THEN
        IPLOT = IPLOT + 1
        WRITE ( OUP, 601 )  IPLOT, RNREV
        CALL PLOT(X,Y,YM,DF,IREC,RMJD,RNREV,IDSAT,IC,NINT,NMAX,
     &            C,F,GX,GY,DGY,DDY,WK,XF,YF,DDT,XA,YA,DYA,IBC,
     &            SIGE,SIGT)
      END IF
C
```

```
 3000 WRITE( OUP, * ) ' '
      WRITE( OUP, * ) ' ', NREC, ' RECORDS IN FILE.'
      WRITE( OUP, * ) ' '
      WRITE( OUP, * ) ' ', IPLOT, ' ELEVATION PROFILES PLOTTED.'
C
      CALL UEND
C
  601 FORMAT(1H0,//,T5,'START PLOT # ',I4,'   OF PASS # ',F6.0)
C
      STOP
      END
C
      SUBROUTINE PLOT ( X, Y, YM, DF, IREC, RMJD, RNREV, IDSAT, IC,
     &                  NINT, NMAX, C, F, GX, GY, DGY, DDY, WK, XF, YF,
     &                  DDT, XA, YA, DYA, IBC, SIGE, SIGT )
C
      REAL*4        X(NMAX), Y(NMAX), YM(NMAX), DF(NMAX), C(NMAX-1,3)
      REAL*4        F(NMAX), GX(NMAX), GY(NMAX), DGY(NMAX), DDY(NMAX)
      REAL*4        WK(7*NMAX+14), XF(NINT+1), YF(NINT+1)
      REAL*4        DDT(NMAX),XA(NMAX),YA(NMAX),DYA(NMAX)
      INTEGER*4     IDSAT
      INTEGER       IBC(NMAX)
      CHARACTER*1   DATE(10)
C
C
C.....EVICT THE BAD DATA POINTS BY COMPARING THE ABSOLUTE DIFFERENCE
C       BETWEEN ADJACENT DATA POINTS
C
      IREC1 = IREC - 1
      REC1 = FLOAT(IREC-1)
C
      DO 110  I = 1, NMAX
        GX(I) = -1.0
  110 CONTINUE
C
      DO 111  I = 1, IREC1
        DDY(I) = ABS( Y(I+1) - Y(I) )
  111 CONTINUE
C
      DO 112  J = 1, IREC1
        IF(DDY(J).LT.SIGE)  THEN
          K = J
          GO TO 222
        END IF
  112 CONTINUE
C
  222 CONTINUE
      DO 113  J = 1, IREC
        IF(DDY(IREC-J).LT.SIGE)  THEN
          IRECG = IREC-J
          GO TO 223
        END IF
  113 CONTINUE
C
  223 KB = 0
      DO 224  I = 1, IRECG-1
        IF(ABS(DDY(I)).GT.SIGE)  KB = KB + 1
  224 CONTINUE
C
      LG = 1
      GX(LG) = X(K)
      GY(LG) = Y(K)
      DGY(LG) = DF(K)
C
  888 LG = LG + 1
      K = K + 1
      IF(ABS(DDY(K-1)).GT.SIGE)  THEN
        K = K + 1
        DO 225  I = 1, KB-1
          IF(ABS(DDY(K-1)).GT.SIGE)  K = K + 1
```

```
  225   CONTINUE
      END IF
      GX(LG) = X(K)
      GY(LG) = Y(K)
      DGY(LG) = DF(K)
      IF(K.LE.IRECG)  GO TO 888
C
C     IB = 0
C     DO 234  K = 1, LG-1
C       IF(DDT(K).GT.SIGT)  THEN
C         IB = IB + 1
C         IBC(IB) = K
C       END IF
C 234 CONTINUE
C
C     IF(IB.EQ.1)  THEN
C       KXG1 = IBC(1)
C       KXG2 = LG
C     END IF
C
C     IF(IB.EQ.1)  LG = KXG1
C
C     NA = 0
C     DO 999  N = 1, LG, 3
C       NA = NA + 1
C       SX = 0.
C       SY= 0.
C       SYA = 0.
C       JC = 0
C       DO 900  L = N, N+2
C         IF(GX(L).LT.0.)  GO TO 246
C         JC = JC + 1
C         SX = SX + GX(L)
C         SY = SY + GY(L)
C         SYA = SYA + DGY(L)
C 900   CONTINUE
C 246   RJC = FLOAT(JC)
C       XA(NA) = SX/RJC
C       YA(NA) = SY/RJC
C       DYA(NA) = SYA/RJC
C 999 CONTINUE
C
C     IF(IB.EQ.1)  THEN
C       LG = KXG2
C       DO 998  N = KXG1+1, LG, 3
C         NA = NA + 1
C         SX = 0.
C         SY = 0.
C         SYA = 0.
C         JC = 0
C         DO 910  L = N, N+2
C           IF(GX(L).LT.0.0)  GO TO 357
C           JC = JC + 1
C           SX = SX + GX(L)
C           SY = SY + GY(L)
C           SYA = SYA + DGY(L)
C 910     CONTINUE
C 357     RJC = FLOAT(JC)
C         XA(NA) = SX/RJC
C         YA(NA) = SY/RJC
C         DYA(NA) = SYA/RJC
C 998   CONTINUE
C     END IF
C
C.....DETERMINE THE CUBIC SPLINE COEFFICIENTS C OF THE ELEVATION PROFILE
C
      IF(LG.GE.8)  THEN
        SM = FLOAT(LG)
        CALL ICSSCU(GX,GY,DGY,LG,SM,F,C,IC,WK,IER)
      END IF
```

```
C
C.....START THE GRAPHIC PROCEDURES
C
      CALL USET('PERCENTUNITS')
      CALL USET('EXTRALARGE')
      CALL UVWPRT(0.0, 99.0, 0.0, 99.0)
      CALL UOUTLN
C
C.....PRINT TITLE, JULIAN DATE AND PASS NUMBER.
C
      CALL USET('CJUST')
      CALL USET('TJUST')
      CALL UPSET('PRECISION',6.0)
      CALL UPRINT(50.,97.,
     &            'ADJUSTED BLACK SEA ALTIMETER ELEVATION PROFILES$')
      CALL UPRINT(50.,93.,'MODIFIED JULIAN DATE : $')
      CALL UMOVE(68.,93.)
      CALL UPRNT1(RMJD,'REAL')
      CALL UPRINT(50.,89.,'PASS : $')
      CALL UMOVE(58.,89.)
      CALL UPRNT1(RNREV,'REAL')
      IF ( IDSAT. EQ. 7502701 )  THEN
        CALL UPRINT(50.,85.,'SATELLITE : GEOS-3$')
      END IF
      IF ( IDSAT. EQ. 7806401 )  THEN
        CALL UPRINT(50.,85.,'SATELLITE : SEASAT$')
      END IF
      CALL ZTIME(DATE,8)
      CALL FMOVE(DATE(10),1,'$')
      CALL UPRINT(88.,6.,DATE)
      CALL UPRINT(88.,3.,'STX/ZMAYA$')
C
C.....DRAW AND LABEL AXES  (DEFAULT TIC MARKS)
C
      CALL USET('NOORIGIN')
      CALL UVWPRT(5.,95.,7.,85.)
C     CALL USET('DSYMBOL')
C     CALL UPSET('SYMBOL',5.0)
C     CALL UPSET('SZSYMBOL',1.0)
      CALL USET('LARGE')
      CALL UPSET('XLABEL','FRACTION OF DAY PAST MIDNIGHTS$')
      CALL UPSET('YLABEL','SURFACE ELEVATION IN METERS$')
      CALL USET('XBOTH')
      CALL USET('YBOTH')
      CALL USET('OWNSCALE')
C     CALL UPSET('TICX',TCX)
C     CALL UPSET('TICY',TCY)
      XMIN = X(1)
      XMAX = X(IREC)
      YMIN = GY(1)
      YMAX = GY(1)
      DO 444  I = 1,LG
        YMIN = AMIN1 ( GY(I), YMIN )
        YMAX = AMAX1 ( GY(I), YMAX )
  444 CONTINUE
      DO 445  I = 1,IREC
        YMIN = AMIN1 ( YM(I), YMIN )
        YMAX = AMAX1 ( YM(I), YMAX )
  445 CONTINUE
C
C.....DETERMINE THE INTERPOLATION INTERVALS AND PERFORM INTERPOLATION
C
      DX = ( GX(LG) - GX(1) ) / FLOAT(NINT)
      XF(1) = GX(1)
      DO 555  I = 2, NINT+1
        XF(I) = XF(I-1) + DX
  555 CONTINUE
C
      IF(LG.GE.8)  THEN
        CALL ICSEVU(GX,F,LG,C,IC,XF,YF,NINT+1,IER)
```

```
      END IF
C
      DO 666  I = 1,NINT+1
        YMIN = AMIN1 ( YF(I), YMIN )
        YMAX = AMAX1 ( YF(I), YMAX )
  666 CONTINUE
C
      SX = 0.020 * ( XMAX - XMIN )
      SY = 0.020 * ( YMAX - YMIN )
      XMIN = XMIN - SX
      XMAX = XMAX + SX
      YMIN = YMIN - SY
      YMAX = YMAX + SY
C     SINDEX = AINT ( ABS(ALOG10(XMIN)) ) + 4.
C     XSCALE = 10.**SINDEX
C     XMIN = AINT(XMIN*XSCALE)/XSCALE
C     XMAX = AINT((XMAX*XSCALE)+1.)/XSCALE
      YMIN = AINT(YMIN) - AMOD(AINT(YMIN),2.)
      YMAX = AINT(YMAX) + 2. - AMOD( AINT(YMAX), 2. )
      CALL UWINDO(XMIN,XMAX,YMIN,YMAX)
      CALL UAXIS(XMIN,XMAX,YMIN,YMAX)
C
C.....PLOT THE DATA POINTS
C
      DO 777 I = 1, LG
        CALL UMOVE(GX(I),GY(I))
        CALL UPRNT1('X$','HORIZ')
  777 CONTINUE
C
C.....PLOT THE SMOOTHED PROFILE
C
      IF(LG.GE.8)  THEN
        PTS = FLOAT(NINT+1)
        CALL USET('LNULL')
        CALL ULINE(XF,YF,PTS)
      END IF
C
C.....PLOT THE CORRESPONDING GEOID MODEL
C
      PTS = FLOAT(IREC)
      CALL USET('DNULL')
      CALL ULINE(X,YM,PTS)
C
C.....DISPLAY FOR SCREEN
C
C     CALL UPAUSE
C
C.....TERMINATED THIS PLOT
C
      CALL UERASE
      CALL URESET
C
      RETURN
      END
//*
//*SYSLIB     DD DSN=SYS2.IMSLD,DISP=SHR
//*           DD DSN=SYS2.IMSLS,DISP=SHR
//*
//GO.FT03F001 DD DSN=ZMAYA.BLACKA.DATA,DISP=SHR
//GO.FT04F001 DD DSN=ZMAYA.ALTIM.DATA(BLKRAPP),DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** **** *
//*                                                              *
//*       BLACK SEA DATA ADJUSTED FOR ORBIT BIAS                 *
//*                                                              *
//* **** **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT06F001 DD SYSOUT=*
// EXEC NOTIFYTS
```

A listing of PROGRAM SORT

```
//ZMAYABS4 JOB (G0109,360,2),AYAU,TIME=(0,30),CLASS=O,MSGCLASS=X
/*JOBPARM LINES=60
// EXEC FORTVC
//SYSIN DD *
C
C FORMAT OF GEOS-3/SEASAT ALTIMETER DATA
C
C VARIABLE    TYPE    DESCRIPTION
C
C     *        I4     SATELITE ID
C              I2     MEASUREMENT TYPE ( 42= OVER LAND , 43 = OVER WATER )
C              I2     TIME SYSTEM ( NM )
C              I4     STATION NUMBER
C              I4     PREPROCESSING INDICATORS
C              I4     MODIFIED JULIAN DATE OF OBSERVATION
C              R8     FRACTION OF DAY PAST MIDNIGHT     (GMT)
C              R8     ALTIMETER OBSERVATION             (METERS)
C     *        I4     SATELLITE GEODETIC LATITUDE       (1E-6 DEGREES)
C     *        I4     SATELLITE EAST LONGITUDE          (1E-6 DEGREES)
C              R4     MEASUREMENT STANDARD DEVIATION    (METERS)
C              I4     NET INSTRUMENT CORRECTION         (MM)
C              I4     METEOROLOGICAL DATA WORD          (GEODYN VOL 3)
C              I4     NET MEDIA CORRECTIONS             (MM)
C              R4     GEOID HEIGHT ABOVE REFERENCE ELLIPSOID (METERS)
C              I4     NET OCEAN DYNAMIC CORRECTIONS     (MM)
C     *        I4     INDICATED SURFACE ELEVATION       (MM)
C     *        I4     S/C REVOLUTION NUMBER
C              I4     MEAN SEA SURFACE ELEVATION (MARSH/MARTIN '81 (MM))
C              I4     DOD REFERENCE RADIAL ORBIT DIFFERENCE (MM)
C              I2     H 1/3                             (CM)
C              I2     AGC                               (DB)
C              I2     WIND SPEED                        (CM/SEC)
C              I2     SURFACE ELEVATION PREPROCESSING WORD
C              I2     DRY TROPOSPHERIC CORRECTION       (MM)
C              I2     FNOC WET TROPOSPHERIC CORRECTION  (MM)
C              I2     SMMR WET TROPOSPHERIC CORRECTION  (MM)
C              I2     IONOSPHERIC CORRECTION            (MM)
C              I2     BAROTROPIC DYNAMIC SEA SURFACE CORRECTION (MM)
C              I2     SOLID EARTH TIDE                  (MM)
C              I2     SCHWIDERSKI OCEAN TIDE            (MM)
C              I2     PARKE OCEAN TIDE                  (MM)
C

      PARAMETER ( ETOP = 50. )
      PARAMETER ( EBOT = 10. )
      PARAMETER ( SIGE = 2. )
      PARAMETER ( MIREC = 150 )
      PARAMETER ( XMIN = 26.5 )
      PARAMETER ( XMAX = 42.5 )
      PARAMETER ( YMIN = 40.0 )
      PARAMETER ( YMAX = 48.0 )
      PARAMETER ( DX = 0.25 )
      PARAMETER ( DY = 0.25 )
      PARAMETER ( CAP = 0.25 )
      PARAMETER ( CAP2 = CAP*CAP )
      PARAMETER ( NXI = (XMAX-XMIN)/DX + 1 )
      PARAMETER ( NYI = (YMAX-YMIN)/DY + 1 )
C
      INTEGER*2  I2,I3,I17,I18,I19,I20,I21,I22,I23,I24,I25,I26,I27,I28
      INTEGER*4  I1,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,I15,I16
      REAL*8     R1,R2
      REAL*4     R3,R4
      REAL*4     GLAT, ELON, HSS, HSUM, HAVG
      REAL*4     GRIDX(NXI), GRIDY(NYI)
      REAL*4     X(MIREC), Y(MIREC), Z(MIREC), ER(MIREC)
      REAL*4     BX(MIREC), BY(MIREC), BZ(MIREC), BER(MIREC), DDZ(MIREC)
      INTEGER    NGD(NXI,NYI)
```

```
      INTEGER    IREC, NREC, NP, NPASS, IDSAT
      INTEGER    NEG, LG, ND, NREV, MG, NTOTAL
      INTEGER    INP, OUP, OUG
C
      DATA   INP / 5 /, OUP / 6 /, OUG / 8 /
      DATA   IREC / 0 /, NREC / 0 /, NEG / 0 /, ND / 0 /, NTOTAL / 0 /
      DATA   NPASS / 1 /
C     DATA   DTR / 3.490658505E-2 /
      DATA   HSUM / 0.0 /
C
C.....FIRST GENERATE THE GRID NET
C
      GRIDX(1) = XMIN
      DO 100  I = 2, NXI
        GRIDX(I) = GRIDX(I-1) + DX
  100 CONTINUE
      GRIDY(1) = YMIN
      DO 101  I = 2, NYI
        GRIDY(I) = GRIDY(I-1) + DY
  101 CONTINUE
C
      READ ( INP )  I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &              I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &              I20,I21,I22,I23,I24,I25,I26,I27,I28
      IDSAT = I1
      NREV = I14
      REWIND INP
C
 1000 READ ( INP, END = 2000 ) I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     *                I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     *                I20,I21,I22,I23,I24,I25,I26,I27,I28
C*****
      IF ( I14. EQ. 12793 )  GO TO 999
C*****
      NREC = NREC + 1
      GLAT = 1.0E-6 * FLOAT(I7)
      ELON = 1.0E-6 * FLOAT(I8)
      HSS  = 1.0E-3 * FLOAT(I13)
C
      IF ( I14. EQ. NREV )  THEN
        IF ( HSS. GT. EBOT. AND. HSS. LT. ETOP )  THEN
          IREC = IREC + 1
          X(IREC) = ELON
          Y(IREC) = GLAT
          Z(IREC) = HSS
          ER(IREC) = R3
        END IF
      ELSE
C
C.....ASSIGN THE PREVIOUS ELEVATION DATA INTO AN ARRAY FOR GRIDDING
C
        IF ( IREC. GT. 4 )  THEN
C
          CALL SELECT( X,Y,Z,ER,IREC, SIGE, LG,BX,BY,BZ,BER,DDZ )
C
          DO 200  I = 1, LG
            ND = ND + 1
            I1 = NINT((BX(I)-CAP-XMIN)/DX)
            J1 = NINT((BY(I)-CAP-YMIN)/DY)
            I2 = NINT((BX(I)+CAP-XMIN)/DX) + 2
            J2 = NINT((BY(I)+CAP-YMIN)/DY) + 2
            IF ( I1. LT. 1 )  I1 = 1
            IF ( J1. LT. 1 )  J1 = 1
            IF ( I2 .GT. NXI )  I2 = NXI
            IF ( J2 .GT. NYI )  J2 = NYI
            DO 201  M = I1, I2
              DO 202  N = J1, J2
                R = (GRIDX(M)-BX(I))**2 + (GRIDY(N)-BY(I))**2
                IF ( R. LE. CAP2 )  THEN
                  NGD(M,N) = NGD(M,N) + 1
```

```
                  WRITE ( OUG )  BX(I),BY(I),BZ(I), M,N, NPASS,BER(I)
                END IF
  202         CONTINUE
  201       CONTINUE
            HSUM = HSUM + BZ(I)
  200     CONTINUE
C
        ELSE
C
          NPASS = NPASS - 1
C
        END IF
C
        IDSAT = I1
        NREV = I14
        NPASS = NPASS + 1
        IREC = 1
        IF ( HSS. GT. EBOT. AND. HSS. LT. ETOP )  THEN
          X(IREC) = ELON
          Y(IREC) = GLAT
          Z(IREC) = HSS
          ER(IREC) = R3
        ELSE
          IREC = IREC - 1
        END IF
      END IF
C
  999 CONTINUE
      GO TO 1000
C
 2000 CONTINUE
C
      IF ( IREC. GT. 4 )  THEN
C
        CALL SELECT( X,Y,Z,ER,IREC, SIGE, LG,BX,BY,BZ,BER,DDZ )
C
          DO 210  I = 1, LG
            ND = ND + 1
            I1 = NINT((BX(I)-CAP-XMIN)/DX)
            J1 = NINT((BY(I)-CAP-YMIN)/DY)
            I2 = NINT((BX(I)+CAP-XMIN)/DX) + 2
            J2 = NINT((BY(I)+CAP-YMIN)/DY) + 2
            IF ( I1. LT. 1 )  I1 = 1
            IF ( J1. LT. 1 )  J1 = 1
            IF ( I2 .GT. NXI )  I2 = NXI
            IF ( J2 .GT. NYI )  J2 = NYI
            DO 211  M = I1, I2
              DO 212  N = J1, J2
                R = (GRIDX(M)-BX(I))**2 + (GRIDY(N)-BY(I))**2
                IF ( R. LE. CAP2 )  THEN
                  NGD(M,N) = NGD(M,N) + 1
                  WRITE ( OUG )  BX(I),BY(I),BZ(I), M,N, NPASS,BER(I)
                END IF
  212         CONTINUE
  211       CONTINUE
            HSUM = HSUM + BZ(I)
  210     CONTINUE
C
      END IF
C
      HAVG = HSUM/FLOAT(ND)
C
      MG = -1
      DO 300  I = 1, NXI
        DO 301  J = 1, NYI
          MG = MAX0( MG, NGD(I,J) )
          NTOTAL = NTOTAL + NGD(I,J)
  301   CONTINUE
  300 CONTINUE
C
```

```
C.....WRITE OUT THE DATA SET ASSOCIATED WITH EACH GRID POINT
C
      DO 400  I = 1, NXI
        DO 401  J = 1, NYI
          NP = NGD(I,J)
          IF ( NP. GT. 0 )  THEN
            NEG = NEG + 1
            WRITE ( OUP, 601 )  I, J, GRIDX(I), GRIDY(J), NP
          END IF
  401   CONTINUE
  400 CONTINUE
C
      WRITE ( OUP, 604 )
      DO 700  I = 1, NXI
        WRITE ( OUP, 605 )  I,(NGD(I,J),J=1,NYI)
  700 CONTINUE
C
      WRITE( OUP, 606 )  NREC, ND, NEG, MG, NTOTAL, HAVG
C
  601 FORMAT(1H0,T5,'GRID (',I2,',',I2,'):',
     &       3X,'ELON = ',F6.2,3X,'GLAT = ',F6.2,
     &       5X,'TOTAL DATA POINTS = ',I6)
C 602 FORMAT((1H ,T5,2F9.4,F12.4))
  603 FORMAT((3(2F7.2,F12.4)))
  604 FORMAT(1H1,///,T20,'DATA DISTRIBUTION MAP:',//)
  605 FORMAT(1H ,T5,I4,5X,33I3)
  606 FORMAT(1H1,////,T5,'# OF RECORDS READ: ',I8,//,
     &       T5,'# OF RECORDS USED IN GRIDDING: ',I8,//,
     &       T5,'# OF NONEMPTY GRID POINTS: ',I4,//,
     &       T5,'MAXIUM # OF DATA PTS TO A GRID PT: ',I5,//,
     &       T5,'TOTAL # OF DATA RECORDS: ',I10,//,
     &       T5,'MEAN GEOID HEIGHT OF THE BLACK SEA: ',F10.4,//,
     &       T5,'GRIDDED DATA HAVE BEEN WRITTEN TO BLACK4B.SORT.')
C
      STOP
      END
      SUBROUTINE SELECT ( X,Y,Z,E,IREC, SIGE, LG,GX,GY,GZ,GE, DDZ )
C
      REAL*4        X(IREC), Y(IREC), Z(IREC), E(IREC)
      REAL*4        GX(IREC), GY(IREC), GZ(IREC), GE(IREC), DDZ(IREC)
C
C.....EVICT THE BAD DATA POINTS BY COMPARING THE ABSOLUTE DIFFERENCE
C       BETWEEN ADJACENT DATA POINTS
C
      IREC1 = IREC - 1
      REC1 = FLOAT(IREC-1)
C
      DO 110  I = 1, IREC
        GX(I) = -1.0
  110 CONTINUE
C
      DO 111  I = 1, IREC1
        DDZ(I) = ABS( Z(I+1) - Z(I) )
  111 CONTINUE
C
      DO 112  J = 1, IREC1
        IF(DDZ(J).LT.SIGE)  THEN
          K = J
          GO TO 222
        END IF
  112 CONTINUE
C
  222 CONTINUE
      DO 113  J = 1, IREC
        IF(DDZ(IREC-J).LT.SIGE)  THEN
          IRECG = IREC-J
          GO TO 223
        END IF
  113 CONTINUE
C
```

```
  223 KB = 0
      DO 224  I = 1, IRECG-1
        IF(ABS(DDZ(I)).GT.SIGE)  KB = KB + 1
  224 CONTINUE
C
      LG = 1
      GX(LG) = X(K)
      GY(LG) = Y(K)
      GZ(LG) = Z(K)
      GE(LG) = E(K)
C
  888 LG = LG + 1
      K = K + 1
      IF(ABS(DDZ(K-1)).GT.SIGE)  THEN
        K = K + 1
        DO 225  I = 1, KB-1
          IF(ABS(DDZ(K-1)).GT.SIGE)  K = K + 1
  225   CONTINUE
      END IF
      GX(LG) = X(K)
      GY(LG) = Y(K)
      GZ(LG) = Z(K)
      GE(LG) = E(K)
      IF(K.LE.IRECG)  GO TO 888
C
      RETURN
      END
//*
// EXEC LINKGOV,REGION=5000K
//*EXEC LINKGOV,REGION.LINK=5000K,REGION.GO=5000K,
//*   PARM='MAP,LIST,SIZE=(4096K,512K)'
//*SYSLIB    DD DSN=SYS2.IMSLS,DISP=SHR
//*          DD DSN=SYS2.IMSLD,DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** *
//*                                                        *
//*           ADJUSTED BLACK SEA DATA                      *
//*                                                        *
//* **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT05F001 DD DSN=ZMAYA.BLACKA.DATA,DISP=SHR
//GO.FT06F001 DD SYSOUT=*
//GO.FT08F001 DD DSN=ZMAYA.BLACK4B.SORT,DISP=SHR
// EXEC NOTIFYTS
```

A listing of PROGRAM WGTAVG

```
//ZMAYABWA JOB (G0109,360,2),AYAU,TIME=(0,30),CLASS=O,MSGCLASS=X
/*JOBPARM LINES=60
// EXEC FORTVC
//SYSIN DD *
C
C FORMAT OF GEOS-3/SEASAT ALTIMETER DATA
C
C VARIABLE     TYPE     DESCRIPTION
C
C      *         I4      SATELITE ID
C                I2      MEASUREMENT TYPE ( 42= OVER LAND , 43 = OVER WATER )
C                I2      TIME SYSTEM ( NM )
C                I4      STATION NUMBER
C                I4      PREPROCESSING INDICATORS
C                I4      MODIFIED JULIAN DATE OF OBSERVATION
C                R8      FRACTION OF DAY PAST MIDNIGHT     (GMT)
C                R8      ALTIMETER OBSERVATION             (METERS)
C      *         I4      SATELLITE GEODETIC LATITUDE       (1E-6 DEGREES)
C      *         I4      SATELLITE EAST LONGITUDE          (1E-6 DEGREES)
C                R4      MEASUREMENT STANDARD DEVIATION    (METERS)
C                I4      NET INSTRUMENT CORRECTION         (MM)
C                I4      METEOROLOGICAL DATA WORD          (GEODYN VOL 3)
C                I4      NET MEDIA CORRECTIONS             (MM)
C                R4      GEOID HEIGHT ABOVE REFERENCE ELLIPSOID (METERS)
C                I4      NET OCEAN DYNAMIC CORRECTIONS     (MM)
C      *         I4      INDICATED SURFACE ELEVATION       (MM)
C      *         I4      S/C REVOLUTION NUMBER
C                I4      MEAN SEA SURFACE ELEVATION (MARSH/MARTIN '81 (MM))
C                I4      DOD REFERENCE RADIAL ORBIT DIFFERENCE (MM)
C                I2      H 1/3                             (CM)
C                I2      AGC                               (DB)
C                I2      WIND SPEED                        (CM/SEC)
C                I2      SURFACE ELEVATION PREPROCESSING WORD
C                I2      DRY TROPOSPHERIC CORRECTION       (MM)
C                I2      FNOC WET TROPOSPHERIC CORRECTION  (MM)
C                I2      SMMR WET TROPOSPHERIC CORRECTION  (MM)
C                I2      IONOSPHERIC CORRECTION            (MM)
C                I2      BAROTROPIC DYNAMIC SEA SURFACE CORRECTION (MM)
C                I2      SOLID EARTH TIDE                  (MM)
C                I2      SCHWIDERSKI OCEAN TIDE            (MM)
C                I2      PARKE OCEAN TIDE                  (MM)
C
C
      PARAMETER ( ETOP = 50. )
      PARAMETER ( EBOT = 10. )
      PARAMETER ( SIGE = 2. )
      PARAMETER ( MIREC = 150 )
      PARAMETER ( XMIN = 26.5 )
      PARAMETER ( XMAX = 42.5 )
      PARAMETER ( YMIN = 40.0 )
      PARAMETER ( YMAX = 48.0 )
      PARAMETER ( DX = 0.25 )
      PARAMETER ( DY = 0.25 )
      PARAMETER ( MXP = 100 )
      PARAMETER ( NXI = (XMAX-XMIN)/DX + 1 )
      PARAMETER ( NYI = (YMAX-YMIN)/DY + 1 )
C
      INTEGER*2  I2,I3,I17,I18,I19,I20,I21,I22,I23,I24,I25,I26,I27,I28
      INTEGER*4  I1,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,I15,I16
      REAL*8     R1,R2
      REAL*4     R3,R4
      REAL*4     GLAT, ELON, HSS, HSUM, HAVG, RSQ, W, WF, GMIN, GMAX
      REAL*4     EX(NXI,NYI,MXP), GY(NXI,NYI,MXP), H(NXI,NYI,MXP)
      REAL*4     GRIDX(NXI), GRIDY(NYI), GRID(NXI,NYI)
      REAL*4     X(MIREC), Y(MIREC), Z(MIREC)
      REAL*4     BX(MIREC), BY(MIREC), BZ(MIREC), DDZ(MIREC)
      INTEGER    NGD(NXI,NYI)
```

```
      INTEGER    IREC, NREC, NP, NEG, LG, ND, NREV, MG
      INTEGER    INP, OUP, OUG
C
      DATA   INP / 5 /, OUP / 6 /, OUG / 8 /
      DATA   IREC / 0 /, NREC / 0 /, NEG / 0 /, ND / 0 /
      DATA   HSUM / 0.0 /, GMIN / 1000.0 /, GMAX / -1000.0 /
C
C.....FIRST GENERATE THE GRID NET
C
      GRIDX(1) = XMIN
      DO 100  I = 2, NXI
        GRIDX(I) = GRIDX(I-1) + DX
  100 CONTINUE
      GRIDY(1) = YMIN
      DO 101  I = 2, NYI
        GRIDY(I) = GRIDY(I-1) + DY
  101 CONTINUE
C
      READ ( INP )  I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     &              I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     &              I20,I21,I22,I23,I24,I25,I26,I27,I28
      NREV = I14
      REWIND INP
C
 1000 READ ( INP, END = 2000 ) I1,I2,I3,I4,I5,I6,R1,R2,I7,I8,R3,
     *                I9,I10,I11,R4,I12,I13,I14,I15,I16,I17,I18,I19,
     *                I20,I21,I22,I23,I24,I25,I26,I27,I28
C*****
      IF ( I14. EQ. 12793 )  GO TO 999
C*****
      NREC = NREC + 1
      GLAT = 1.0E-6 * FLOAT(I7)
      ELON = 1.0E-6 * FLOAT(I8)
      HSS  = 1.0E-3 * FLOAT(I13)
C
      IF ( I14. EQ. NREV )  THEN
        IF ( HSS. GT. EBOT. AND. HSS. LT. ETOP )  THEN
          IREC = IREC + 1
          X(IREC) = ELON
          Y(IREC) = GLAT
          Z(IREC) = HSS
        END IF
      ELSE
C
C.....ASSIGN THE PREVIOUS ELEVATION DATA INTO AN ARRAY FOR GRIDDING
C
        IF ( IREC. GT. 4 )  THEN
C
          CALL SELECT( X,Y,Z,IREC, SIGE, LG,BX,BY,BZ,DDZ )
C
          DO 200  I = 1, LG
            ND = ND + 1
            LX = NINT((BX(I)-XMIN)/DX) + 1
            LY = NINT((BY(I)-YMIN)/DY) + 1
            NGD(LX,LY) = NGD(LX,LY) + 1
            EX(LX,LY,NGD(LX,LY)) = BX(I)
            GY(LX,LY,NGD(LX,LY)) = BY(I)
            H(LX,LY,NGD(LX,LY)) = BZ(I)
            HSUM = HSUM + BZ(I)
  200     CONTINUE
C
        END IF
C
        NREV = I14
        IREC = 1
        IF ( HSS. GT. EBOT. AND. HSS. LT. ETOP )  THEN
          X(IREC) = ELON
          Y(IREC) = GLAT
          Z(IREC) = HSS
        ELSE
```

```
          IREC = IREC - 1
        END IF
      END IF
C
  999 CONTINUE
      GO TO 1000
C
 2000 CONTINUE
C
      IF ( IREC. GT. 4 )  THEN
C
        CALL SELECT( X,Y,Z,IREC, SIGE, LG,BX,BY,BZ,DDZ )
C
          DO 201  I = 1, LG
            ND = ND + 1
            LX = NINT((BX(I)-XMIN)/DX) + 1
            LY = NINT((BY(I)-YMIN)/DY) + 1
            NGD(LX,LY) = NGD(LX,LY) + 1
            EX(LX,LY,NGD(LX,LY)) = BX(I)
            GY(LX,LY,NGD(LX,LY)) = BY(I)
            H(LX,LY,NGD(LX,LY)) = BZ(I)
            HSUM = HSUM + BZ(I)
  201     CONTINUE
C
      END IF
C
      HAVG = HSUM/FLOAT(ND)
C
      MG = -1
      DO 300  I = 1, NXI
        DO 301  J = 1, NYI
          MG = MAX0( MG, NGD(I,J) )
  301   CONTINUE
  300 CONTINUE
C
C.....WRITE OUT THE DATA SET ASSOCIATED WITH EACH GRID POINT
C
      DO 400  I = 1, NXI
        DO 401  J = 1, NYI
          NP = NGD(I,J)
          GRID(I,J) = -1000.0
          IF ( NP. GT. 0 )  THEN
            IF ( NP. EQ. 1 )  THEN
              GRID(I,J) = H(I,J,1)
            ELSE
              WF = 0.0
              W = 0.0
              DO 402  K = 1, NP
                RSQ = (EX(I,J,K)-GRIDX(I))**2 + (GY(I,J,K)-GRIDY(J))**2
                RSQ = 1.0/RSQ
                WF = WF + RSQ*H(I,J,K)
                W = W + RSQ
  402         CONTINUE
              GRID(I,J) = WF/W
            END IF
            GMIN = AMIN1 ( GMIN, GRID(I,J) )
            GMAX = AMAX1 ( GMAX, GRID(I,J) )
            NEG = NEG + 1
            WRITE ( OUP, 601 )  I, J, GRIDX(I), GRIDY(J), GRID(I,J), NP
            WRITE ( OUP, 602 )  (EX(I,J,K),GY(I,J,K),H(I,J,K),K=1,NP)
          END IF
  401   CONTINUE
  400 CONTINUE
C
      DO 700  I = 1, NXI
        WRITE ( OUG, 603 )  (GRIDX(I),GRIDY(J),GRID(I,J),J=1,NYI)
  700 CONTINUE
C
      WRITE ( OUP, 604 )
      DO 800  I = 1, NXI
```

```
        WRITE ( OUP, 605 )  I,(NGD(I,J),J=1,NYI)
  800 CONTINUE
C
      WRITE( OUP, 606 )  NREC, ND, NEG, MG, GMIN, GMAX, HAVG
C
  601 FORMAT(1H0,//,T5,'GRID (',I2,',',I2,'):',
     &       3X,'ELON = ',F6.2,3X,'GLAT = ',F6.2,
     &       3X,'WT. H = ',F10.4,5X,'TOTAL DATA POINTS = ',I3/)
  602 FORMAT((1H ,T5,2F9.4,F12.4))
  603 FORMAT((3(2F7.2,F12.4)))
  604 FORMAT(1H1,///,T20,'DATA DISTRIBUTION MAP:',//)
  605 FORMAT(1H ,T5,I4,5X,33I3)
  606 FORMAT(1H1,////,T5,'# OF RECORDS READ: ',I8,//,
     &       T5,'# OF RECORDS USED IN GRIDDING: ',I8,//,
     &       T5,'# OF NONEMPTY GRID POINTS: ',I4,//,
     &       T5,'MAXIUM # OF DATA PTS TO A GRID PT: ',I3,//,
     &       T5,'THE RANGE OF GEOID HEIGHT IS: ',2F14.4,//,
     &       T5,'MEAN GEOID HEIGHT OF THE BLACK SEA: ',F10.4,//,
     &       T5,'GRIDDED DATA HAVE BEEN WRITTEN TO BLACKA.GRID.')
C
      STOP
      END
      SUBROUTINE SELECT ( X, Y, Z, IREC, SIGE, LG, GX, GY, GZ, DDZ )
C
      REAL*4        X(IREC), Y(IREC), Z(IREC)
      REAL*4        GX(IREC), GY(IREC), GZ(IREC), DDZ(IREC)
C
C.....EVICT THE BAD DATA POINTS BY COMPARING THE ABSOLUTE DIFFERENCE
C       BETWEEN ADJACENT DATA POINTS
C
      IREC1 = IREC - 1
      REC1 = FLOAT(IREC-1)
C
      DO 110  I = 1, IREC
        GX(I) = -1.0
  110 CONTINUE
C
      DO 111  I = 1, IREC1
        DDZ(I) = ABS( Z(I+1) - Z(I) )
  111 CONTINUE
C
      DO 112  J = 1, IREC1
        IF(DDZ(J).LT.SIGE)  THEN
          K = J
          GO TO 222
        END IF
  112 CONTINUE
C
  222 CONTINUE
      DO 113  J = 1, IREC
        IF(DDZ(IREC-J).LT.SIGE)  THEN
          IRECG = IREC-J
          GO TO 223
        END IF
  113 CONTINUE
C
  223 KB = 0
      DO 224  I = 1, IRECG-1
        IF(ABS(DDZ(I)).GT.SIGE)  KB = KB + 1
  224 CONTINUE
C
      LG = 1
      GX(LG) = X(K)
      GY(LG) = Y(K)
      GZ(LG) = Z(K)
C
  888 LG = LG + 1
      K = K + 1
      IF(ABS(DDZ(K-1)).GT.SIGE)  THEN
        K = K + 1
```

```
        DO 225  I = 1, KB-1
          IF(ABS(DDZ(K-1)).GT.SIGE)  K = K + 1
  225   CONTINUE
      END IF
      GX(LG) = X(K)
      GY(LG) = Y(K)
      GZ(LG) = Z(K)
      IF(K.LE.IRECG)  GO TO 888
C
      RETURN
      END
//*
// EXEC LINKGOV,REGION=5000K
//*SYSLIB   DD DSN=SYS2.IMSLS,DISP=SHR
//*         DD DSN=SYS2.IMSLD,DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** *
//*                                                        *
//*           ADJUSTED BLACK SEA DATA                      *
//*                                                        *
//* **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT05F001 DD DSN=ZMAYA.BLACKA.DATA,DISP=SHR
//GO.FT06F001 DD SYSOUT=*
//GO.FT08F001 DD DSN=ZMAYA.BLACKA.GRID,DISP=SHR
// EXEC NOTIFYTS
```

A listing of PROGRAM CONTOUR

```
//ZMAYABGA JOB (G0109,360,10),AYAU,TIME=(1,00),CLASS=O,MSGCLASS=X
//TEMPLATE EXEC G38PLOT
//FORT.SYSIN DD *
C
      PARAMETER ( BIAS =  0.0 )
      PARAMETER ( XMIN = 26.5 )
      PARAMETER ( XMAX = 42.5 )
      PARAMETER ( YMIN = 40.0 )
      PARAMETER ( YMAX = 48.0 )
      PARAMETER ( DX = 0.25 )
      PARAMETER ( DY = 0.25 )
      PARAMETER ( CHGH =  50. )
      PARAMETER ( CLOW =  10. )
      PARAMETER ( CINC =   1. )
      PARAMETER ( CLAB =   2. )
      PARAMETER ( NLON = ( (XMAX-XMIN)/DX + 1 ) )
      PARAMETER ( NLAT = ( (YMAX-YMIN)/DY + 1 ) )
C
      INTEGER       INP, NRU, OUP
      INTEGER       NPROJ, LINEH, LINEV
      INTEGER       LIST(6)
      INTEGER       MASK(NLON,NLAT)
      REAL*4        LON(NLON), LAT(NLAT), GRID(NLON,NLAT)
      REAL*4        WKCNTR(NLON,NLAT), XP(NLON), YP(NLAT)
      REAL*4        PENS(7)
      REAL*4        PLAT,PLONG,VLAT1,VLAT2,VLONG1,VLONG2
      CHARACTER*1   DATE(10)
C
      DATA  INP / 5 /, OUP / 6 /, NRU / 0 /
      DATA  GMAX / -999.0 /, GMIN / 1000.0 /
      DATA  PENS / 7*4.0 /
      DATA  LIST / 8, 110, 75, 0, 7, 10 /
      DATA  NPROJ / 16 /
      DATA  PLAT / 90. /, PLONG / 0. /
C
      DO 100  I = 1, NLON
        READ ( INP, 510, END = 777 )  (LON(I),LAT(J),GRID(I,J),J=1,NLAT)
  100 CONTINUE
C
  777 CONTINUE
C
      DO 111  I = 1, NLON
        DO 112  J = 1, NLAT
          IF ( GRID(I,J). NE. -1000.0 )  THEN
            NRU = NRU + 1
            GRID(I,J) = GRID(I,J) - BIAS
            GMAX = AMAX1 ( GRID(I,J), GMAX )
            GMIN = AMIN1 ( GRID(I,J), GMIN )
          END IF
  112   CONTINUE
  111 CONTINUE
C
      WRITE( OUP, 610 )  GMIN, GMAX, NRU
C
      WRITE(OUP,620) ((LON(IX),LAT(IY),GRID(IX,IY),IY=1,NLAT),IX=1,NLON)
C
C.....START GRAPHIC PROCEDURES
C
      CALL USTART
      CALL UINQES(1.0,SUPRT)
      IF (SUPRT.NE.0.0)  CALL UESCAP(1.0,LIST,6.0,IDUM,DUM)
      CALL USET('PERCENTUNITS')
      CALL UVWPRT(0.0,99.0,0.0,99.0)
      CALL UOUTLN
C
C.....PUT ON LABELS
C
```

```
      CALL USET('CJUST')
      CALL USET('TJUST')
      CALL USET('YELLOW')
      CALL USET('SOFTWARE')
      CALL USET('EXTRALARGE')
      CALL UPSET('VERTICAL SIZE',2.0)
      CALL UPSET('HORIZONT SIZE',1.6)
      CALL UFONT('TROM')
      CALL UPRINT(50.,96.,
     &            'WT. AVG. GEOIDAL HEIGHT OF THE BLACK SEA$')
C
      VLAT1 = YMIN
      VLAT2 = YMAX
      VLONG1 = XMIN
      VLONG2 = XMAX
      LINEH = IFIX(YMAX-YMIN)
      LINEV = IFIX(XMAX-XMIN)
      CALL WDMAP(NPROJ,PLAT,PLONG,VLAT1,VLAT2,VLONG1,VLONG2,
     &  LINEH,LINEV)
C
C.....LABEL AXIS AND WRITE LON AND LATS ON PLOT
C
      DLONG = ABS(VLONG2 - VLONG1) / LINEV
      LINES = LINEV + 1
      DIX = 100.0 / LINEV
      YLONGI = -3.5
      DO 420 I = 1, LINES, 2
        V = VLONG1 + (DLONG * (I - 1))
        XLATIT = DIX * (I - 1)
        CALL UMOVE(XLATIT,YLONGI)
        CALL UPRNT1(V,'REAL')
  420 CONTINUE
C
C.....WRITE LABEL
C
      CALL USET('BJUS')
      CALL UMOVE(50.,-7.5)
      CALL UPRNT1('LONGITUDE$','HORIZ')
C
      DLAT = ABS(VLAT2 - VLAT1) / LINEH
      LINES = LINEH + 1
      DIY = 100.0 / LINEH
      XLT = -3.5
      DO 430 I = 1, LINES, 2
        V = VLAT1 + (DLAT * (I - 1))
        YLO = DIY * (I - 1)
        CALL UMOVE(XLT,YLO)
        CALL UPRNT1(V,'REAL')
  430 CONTINUE
C
C.....WRITE LABEL
C
      CALL USET('MJUS')
      CALL UMOVE(-7.5,50.0)
      CALL UPRNT1('LATITUDE$','VERTI')
C
C.....WRITE DATE AND ID ON PLOT
C
      CALL ZTIME(DATE,8)
      CALL FMOVE(DATE(10),1,'$')
      CALL UPRINT(78.,-8.,DATE)
      CALL UPRINT(95.,-8.,'STX/ZMAYA$')
C
C.....START CONTOURING PROCEDURES
C
      CALL USET('NOMI')
      CALL USET('SMOOTH')
      CALL UPSET('CLOWEST', CLOW)
      CALL UPSET('CINCREMENT' ,CINC)
      CALL UPSET('CHIGHEST', CHGH)
```

```
      CALL UPSET('CLABEL', CLAB)
C
      CALL USET('NOAXES')
      CALL USET('NOXLABEL')
      CALL USET('NOYLABEL')
C
C.....CONVERT LAT/LON TO PLOT COORDS.
C
      DO 222  I=1,NLON
        CALL WDPOS(1.0,LON(I),XP(I),DUM)
  222 CONTINUE
      DO 223  J = 1, NLAT
        CALL WDPOS(LAT(J),1.0,DUM,YP(J))
  223 CONTINUE
C
C     WRITE(OUP,630)
C     WRITE(OUP,620) ((XP(IX),YP(IY),GRID(IX,IY),IY=1,NLAT),IX=1,NLON)
C
      FX = FLOAT(NLON)
      FY = FLOAT(NLAT)
      CALL UVWPRT(0.0,99.0,0.0,99.0)
      CALL USET('OWNSCALE')
      CALL UPSET('EXCEPTION',-1000.0)
      CALL USET('DNULL')
      CALL UPCNTR(GRID,XP,YP,WKCNTR,FX,FY,PENS)
C
C     CALL UPAUSE
      CALL UEND
C
C
  510 FORMAT((3(2F7.2,F12.4)))
  610 FORMAT(1H1,T5,'THE RANGE OF GRID VALUES IS FROM ', F12.4,
     & ' TO ',F12.4,' IN ',I7,' UNMASKED RECORDS.'//,
     & T10,'LONGITUDE, LATITUDE AND SURFACE HEIGHT',/)
  620 FORMAT((1H ,4(2F7.2,2X,F12.4,1X)))
  630 FORMAT(1H1,T15,'COORDINATES AND GRID VALUES FOR CONTOURING',/)
C
      STOP
      END
//*
//GO.FT05F001 DD DSN=ZMAYA.BLACKA.GRID,DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** *
//*                                                        *
//*    ADJUSTED BLACK SEA GRID DATA USING WT. AVERAGING    *
//*                                                        *
//* **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT06F001 DD SYSOUT=*
//GO.FT19F001 DD DSN=SYS2.WRLDATA2,DISP=SHR,LABEL=(,,,IN)
//*
// EXEC NOTIFYTS
```

A listing of PROGRAM GEOID

```
//ZMAYABSH JOB (G0109,360,2),AYAU,TIME=(2,30),CLASS=O,MSGCLASS=X
// EXEC FORTVC
//SYSIN DD *
C
      PARAMETER ( MPASS = 142 )
      PARAMETER ( NREFP = 73 )
      PARAMETER ( KUTOFF = 3 )
      PARAMETER ( LCTV = 1 )
      PARAMETER ( MTR = 160 )
      PARAMETER ( XMIN = 26.5 )
      PARAMETER ( XMAX = 42.5 )
      PARAMETER ( YMIN = 40.0 )
      PARAMETER ( YMAX = 48.0 )
      PARAMETER ( DX = 0.25 )
      PARAMETER ( DY = 0.25 )
      PARAMETER ( LCT = 33 )
      PARAMETER ( NXI = (XMAX-XMIN)/DX + 1 )
      PARAMETER ( NYI = (YMAX-YMIN)/DY + 1 )
      PARAMETER ( LEN = (MTR*(MTR+1))/2 )
      PARAMETER ( MSYM = (MPASS*(MPASS-1))/2 )
C
      REAL*4      SE(MTR)
      REAL*4      GRIDX(NXI), GRIDY(NYI)
      REAL*4      GRID(NXI,NYI), VARM(NXI,NYI)
      REAL*8      RGRIDX(NXI), RGRIDY(NYI)
      REAL*8      SY(MTR), CY(MTR)
      REAL*8      EX(MTR), GY(MTR), GH(MTR), SGY(MTR), CGY(MTR)
      REAL*8      CM(LEN), CMI(LEN), CMS(LEN), DM(LEN), DLL(LEN)
      REAL*8      CP(MTR), SRC(MTR), AP(MTR)
      REAL*8      SD(LCT), CTV(LCT,3), CV(LCT), SC(LCT-1,3)
      REAL*8      DTR, ELON, GLAT, X1, Y1, SY1, CY1, PVAL, VAR
      INTEGER     IDIAG(MTR), NPASS(MTR)
      INTEGER     INP, OUP, OUG, OUH, OUC, OUE
C
      DATA    INP / 5 /, OUP / 6 /, OUG / 8 /, OUH / 9 /
      DATA    OUC / 10 /, OUE / 11 /
      DATA    NIER / 0 /
      DATA    DTR / 3.490658504D-2 /
      DATA    GMIN / 1000.0 /, GMAX / -999.0 /
      DATA    VMIN / 1000.0 /, VMAX / -999.0 /
C
C
C.....GENERATE A GRID NET
C
      GRIDX(1) = XMIN
      DO 100   I = 2, NXI
        GRIDX(I) = GRIDX(I-1) + DX
        RGRIDX(I) = DBLE(GRIDX(I)) * DTR
  100 CONTINUE
      GRIDY(1) = YMIN
      DO 101   I = 2, NYI
        GRIDY(I) = GRIDY(I-1) + DY
        RGRIDY(I) = DBLE(GRIDY(I)) * DTR
        SY(I) = DSIN(RGRIDY(I))
        CY(I) = DCOS(RGRIDY(I))
  101 CONTINUE
      DO 102   I = 1, NXI
        DO 103   J = 1, NYI
          GRID(I,J) = -1000.0
          VARM(I,J) = -1000.0
  103   CONTINUE
  102 CONTINUE
C
C.....READ IN THE COVARIANCE TABLE
C
      DO 110   I = 1, LCT
        READ ( INP, 501 )   SD(I), (CTV(I,J),J=1,3)
```

```
          CV(I) = CTV(I,LCTV)
  110 CONTINUE
C
C.....DETERMINE A CUBIC SPLINE COEFFICIENT MATRIX SC
C
      ISC = LCT - 1
      CALL ICSCCU ( SD, CV, LCT, SC, ISC, IER )
C
C.....WRITE OUT THE VALUES OF THE COVARIANCE TABLE USED
C
      WRITE ( OUP, 601 )
      WRITE ( OUP, 602 )  (SD(I),CTV(I,LCTV),I=1,LCT)
C
      WRITE ( OUP, 600 )
C
C.....READ IN THE ERROR COVARIANCE MATRIX FROM PASS BIAS ADJUSTMENT
C
      READ ( OUE, 502 )  ( DM(M),M=1,MPASS1 )
C
C.....CONSTRUCT THE COVARIANCE MATRIX DIAGONAL ADDRESS
C
      DO 120  I = 1, MTR
        IDIAG(I) = (I*(I+1))/2
  120 CONTINUE
C
C.....READ IN THE SORTED DATA
C
      READ ( OUG )  R1, R2, R3, I1, I2, I3, R4
      IKEEP = I1
      JKEEP = I2
      REWIND OUG
C
 1000 READ ( OUG, END = 2000 )  R1, R2, R3, I1, I2, I3, R4
      ELON = DBLE(R1) * DTR
      GLAT = DBLE(R2) * DTR
      HSS = R3
C
      IF ( I1. EQ. IKEEP. AND. I2. EQ. JKEEP )  THEN
C
        IREC = IREC + 1
        EX(IREC) = ELON
        GY(IREC) = GLAT
        GH(IREC) = DBLE(HSS)
        NPASS(IREC) = I3
        SE(IREC) = R4
        IF ( NPASS(IREC). EQ. NREFP )  NPASS(IREC) = 0
        IF ( NPASS(IREC). GT. NREFP )  NPASS(IREC) = NPASS(IREC) - 1
C
      ELSE
C
        IF ( IREC. GT. KUTOFF )  THEN
          X1 = RGRIDX(IKEEP)
          Y1 = RGRIDY(JKEEP)
          SY1 = SY(JKEEP)
          CY1 = CY(JKEEP)
          JREC = (IREC*(IREC+1))/2
          JC = JC + 1
C         IF ( JC.GE.571 .AND. JC.LE.590 )  THEN
C           WRITE ( OUP, * )  ' '
C           WRITE ( OUP, 678 )  (EX(LK),GY(LK),GH(LK),NPASS(LK),SE(LK),
C    *                                 LK=1,IREC)
C 678       FORMAT((1H ,3F12.4,I5,F12.4))
C           WRITE ( OUP, * )  ' '
C         END IF
          CALL PREDIC ( PVAL,VAR, EX,GY,GH,SGY,CGY,IREC,
     *                  NPASS, SE, DM, DLL, X1,Y1,SY1,CY1,
     *                  CM,CMS,CMI,JREC, IDIAG,CP,SRC,AP,
     *                  SD,CTV,CV,SC,LCT,ISC,
     *                  LCTV,DTR, IER, NIER )
          IF ( IER. EQ. 0 )  THEN
```

```
              GRID(IKEEP,JKEEP) = PVAL
              VARM(IKEEP,JKEEP) = VAR
            ELSE
              JC = JC - 1
              WRITE ( OUP, * )  'IER =',IER,'      PVAL =',PVAL,
     *                                              VAR =',VAR
            END IF
            WRITE ( OUP, 603 )  IKEEP,JKEEP,GRIDX(IKEEP),GRIDY(JKEEP),
     *             GRID(IKEEP,JKEEP), VARM(IKEEP,JKEEP), IREC,JC
          END IF
C
          IKEEP = I1
          JKEEP = I2
          IREC = 1
          EX(IREC) = ELON
          GY(IREC) = GLAT
          GH(IREC) = DBLE(HSS)
          NPASS(IREC) = I3
          SE(IREC) = R4
          IF ( NPASS(IREC). EQ. NREFP )  NPASS(IREC) = 0
          IF ( NPASS(IREC). GT. NREFP )  NPASS(IREC) = NPASS(IREC) - 1
C
        END IF
C
        GO TO 1000
C
 2000 CONTINUE
C
        IF ( IREC. GT. KUTOFF )  THEN
          X1 = RGRIDX(I1)
          Y1 = RGRIDY(I2)
          SY1 = SY(JKEEP)
          CY1 = CY(JKEEP)
          JREC = (IREC*(IREC+1))/2
          JC = JC + 1
          CALL PREDIC ( PVAL,VAR, EX,GY,GH,SGY,CGY,IREC,
     *                  NPASS, SE, DM, DLL, X1,Y1,SY1,CY1,
     *                  CM,CMS,CMI,JREC, IDIAG,CP,SRC,AP,
     *                  SD,CTV,CV,SC,LCT,ISC,
     *                  LCTV,DTR, IER, NIER )
          IF ( IER. EQ. 0 )  THEN
            GRID(I1,I2) = PVAL
            VARM(I1,I2) = VAR
          ELSE
            JC = JC - 1
            WRITE ( OUP, * )  'IER =',IER,'      PVAL =',PVAL,
     *                                            VAR =',VAR
          END IF
          WRITE ( OUP, 603 )  I1,I2,GRIDX(I1),GRIDY(I2),
     *                        GRID(I1,I2), VARM(I1,I2), IREC,JC
        END IF
C
C.....DETERMINE THE RANGE OF THE PREDICTED VALUES AND THEIR VARIANCE
C
        DO 300  I = 1, NXI
          DO 301  J = 1, NYI
            IF ( GRID(I,J). NE. -1000.0 )  THEN
              GMIN = AMIN1 ( GMIN, GRID(I,J) )
              GMAX = AMAX1 ( GMAX, GRID(I,J) )
            END IF
            IF ( VARM(I,J). NE. -1000.0 )  THEN
              VMIN = AMIN1 ( VMIN, VARM(I,J) )
              VMAX = AMAX1 ( VMAX, VARM(I,J) )
            END IF
  301     CONTINUE
  300   CONTINUE
C
C.....PRINT OUT THE GRIDDED VALUES AND THE VARIANCE
C
        NXY = NXI*NYI
```

```
      WRITE ( OUP, 604 )  NIER, JC, NXY, GMIN, GMAX
      DO 400  I = 1, NXI
        WRITE ( OUP, 605 )  (GRIDX(I),GRIDY(J),GRID(I,J),J=1,NYI)
        WRITE ( OUH, 606 )  (GRIDX(I),GRIDY(J),GRID(I,J),J=1,NYI)
  400 CONTINUE
C
      WRITE ( OUP, 607 )  VMIN, VMAX
      DO 410  I = 1, NXI
        WRITE ( OUP, 605 )  (GRIDX(I),GRIDY(J),VARM(I,J),J=1,NYI)
        WRITE ( OUC, 606 )  (GRIDX(I),GRIDY(J),VARM(I,J),J=1,NYI)
  410 CONTINUE
C
  501 FORMAT(1X,D10.4,3D20.13)
  502 FORMAT((4D20.13))
  600 FORMAT(1H1)
  601 FORMAT(1H1,T5,'VALUES OF THE COVARIANCE TABLE USED:',//)
  602 FORMAT((1H ,5X,D10.4,5X,D20.13))
  603 FORMAT(1H ,T2,'GRID(',I2,',',I2,')',3X,'ELON = ',F6.2,2X,
     *       ',GLAT = ',F6.2,5X,'GEOID HEIGHT = ',F10.4,' +- ',F8.2,5X,
     *       'RECORD #: ',I5,5X,'COUNTER: ',I5,/)
  604 FORMAT(1H1,T5,I4,' GRID POINTS HAVING NEGATIVE VARIANCE',//,
     *           T5,'THERE ARE ',I4,' POINTS GRIDDED OUT OF A NET ',
     *              ' OF ',I5,' GRID POINTS.',//,T8,'THE RANGE OF THE',
     *              ' GEODIAL HEIGHT IN METERS IS:',2F12.4,//)
  605 FORMAT((1H ,5(2F7.2,F12.4)))
  606 FORMAT((3(2F7.2,F12.4)))
  607 FORMAT(1H1,T5,'THE RANGE OF THE VARIANCE IN CM IS:',2F12.4,//)
C
      STOP
      END
      SUBROUTINE PREDIC ( PVAL,VAR, EX,GY,GH, SGY,CGY, IREC,
     *                    NPASS, SE, DM, DLL, X1,Y1,SY1,CY1,
     *                    CM,CMS,CMI,JREC, IDIAG,CP,SRC,AP,
     *                    SD,CTV,CV,SC,LCT,ISC,
     *                    LCTV,DTR, IER, NIER )
C
      REAL*4      SE(IREC)
      REAL*8      EX(IREC), GY(IREC), GH(IREC), SGY(IREC), CGY(IREC)
      REAL*8      CP(IREC), SRC(IREC), AP(IREC)
      REAL*8      CM(JREC), CMI(JREC), CMS(JREC), DM(JREC), DLL(JREC)
      REAL*8      SD(LCT), CTV(LCT,3), CV(LCT), SC(LCT-1,3)
      REAL*8      D1, D2, CSD, DSD, VDSD, AVG, DTR
      REAL*8      X1, Y1, SY1, CY1, PVAL, VAR
      INTEGER     IDIAG(IREC), NPASS(IREC)
C
C.....RESET THE ERROR COUNTER
C
      IER = 0
C
C.....CLEAR THE COVARIANCE MATRIX ARRAY
C
      CALL DCLR( CM, JREC )
      CALL DCLR( CMI, JREC )
      CALL DCLR( CMS, JREC )
      CALL DCLR( DLL, JREC )
C
C.....DETERMINE THE AVERAGE VALUE OF THE DATA SET
C
      AVG = 0.0
      DO 100  I = 1, IREC
        AVG = AVG + GH(I)
  100 CONTINUE
      AVG = AVG / DFLOAT(IREC)
C
C.....CONSTRUCT THE DIAGONAL TERMS OF THE SIGNAL COVARIANCE MATRIX,
C       REMOVE THE AVERAGE VALUE FROM DATA SET TO MAKE IT ZERO MEAN
C       AND SET UP THE DATA VECTOR IN DOUBLE PRECISION
C
      DO 110  I = 1, IREC
        CM(IDIAG(I)) = CTV(1,LCTV) + DBLE(SE(I)*SE(I))
```

```
          GH(I) = GH(I) - AVG
  110 CONTINUE
C
C.....EVALUATE ALL THE SINE AND COSINE TERMS
C
      DO 200  I = 1, IREC
        SGY(I) = DSIN(GY(I))
        CGY(I) = DCOS(GY(I))
  200 CONTINUE
C
C.....CONSTRUCT THE OFF-DIAGONAL TERMS OF THE SIGNAL COVARIANCE MATRIX
C
      DO 300  J = 2, IREC
        DO 301  I = 1, J-1
C
          CSD = SGY(I)*SGY(J) + CGY(I)*CGY(J)*DCOS(EX(I)-EX(J))
          DSD = DACOS(CSD)/DTR
          CALL ICSEVU ( SD, CV, LCT, SC, ISC, DSD, VDSD, 1, IER )
          K = (J*(J-1))/2 + I
          CM(K) = VDSD
C
  301   CONTINUE
  300 CONTINUE
C
C.....ASSEMBLE THE ERROR COVARIANCE MATRIX FROM BIAS ADJUSTMENT
C
      DO 310  J = 1, IREC
        DO 311  I = 1, J
C
          K = (J*(J-1))/2 + I
          IF ( NPASS(I). NE. 0. AND. NPASS(J). NE. 0 )  THEN
            M1 = MIN0 ( NPASS(I), NPASS(J) )
            M2 = MAX0 ( NPASS(I), NPASS(J) )
            L = (M2*(M2-1))/2 + M1
            DLL(K) = DM(L)
          ELSE
            DLL(K) = 0.0D0
          END IF
C
  311   CONTINUE
  310 CONTINUE
C
C.....ASSEMBLE THE COMPLETE COVARIANCE MATRIX AND
C.....  DUPLICATE THE COVARIANCE MATRIX FOR STORAGE
C
      DO 320  I = 1, JREC
        CM(I) = CM(I) + DLL(I)
        CMS(I) = CM(I)
  320 CONTINUE
C
C.....INVERT THE COVARIANCE MATRIX
C
      CALL LINV1P ( CMS,IREC,CMI, IDGT, D1,D2, IER )
C     CALL SOLVE ( CMS, SRC, IDIAG, IREC, .TRUE., .FALSE. )
C     CALL SYMINV ( CMS, SRC, IDIAG, IREC, CMI )
C
      IF ( IER. EQ. 0 )  THEN
C
C.......CONSTRUCT THE COVARIANCE VECTOR
C
        DO 330  I = 1, IREC
C
          CSD = SY1*SGY(I) + CY1*CGY(I)*DCOS(X1-EX(I))
          DSD = DACOS(CSD)/DTR
          CALL ICSEVU ( SD, CV, LCT, SC, ISC, DSD, VDSD, 1, IER )
          CP(I) = VDSD
C
  330   CONTINUE
C
C.......DETERMINE THE PREDICTED VALUE AT A GRID POINT
```

```
C
          CALL VMULSF ( CMI,IREC, GH,1,IREC, SRC,IREC )
C         CALL DCLR ( SRC, IREC )
C         CALL VSMXVT ( CMI, GH, SRC, IDIAG, IREC )
          PVAL = DOT ( CP, SRC, IREC ) + AVG
C
C.......DETERMINE THE VARIANCE OF THE PREDICTED VALUE
C
          CALL VMULSF ( CMI,IREC, CP,1,IREC, AP,IREC )
          CALL VMULSF ( CM,IREC, AP,1,IREC, SRC,IREC )
C         CALL DCLR ( AP, IREC )
C         CALL DCLR ( SRC, IREC )
C         CALL VSMXVT ( CMI, CP, AP, IDIAG, IREC )
C         CALL VSMXVT ( CM, AP, SRC, IDIAG, IREC )
          VAR = CTV(1,LCTV) - 2.0D0*DOT(AP,CP,IREC) + DOT(AP,SRC,IREC)
C
          IF ( VAR. GT. 0.0 )  THEN
            VAR = 100.0D0*DSQRT(VAR)
          ELSE
            IER = 1
            NIER = NIER + 1
          END IF
C
        END IF
C
        RETURN
        END
        FUNCTION DOT (A,B,N)
C
C  Compute the dot product of the two N-vectors A and B.
C
        INTEGER  N, I
        REAL*4   DOT
        REAL*8   A(1),B(1)
C
        DOT=0.0
C
        DO 100  I=1,N
  100   DOT=DOT + A(I)*B(I)
C
        RETURN
        END
        SUBROUTINE SYMINV ( A, B, IDIAG, NEQ, AINV )
C
        IMPLICIT REAL*8  (A-H,O-Z)
        DIMENSION     A(1), B(1), AINV(1), IDIAG(1)
C
        K = 0
        DO 100  I = 1, NEQ
          DO 101  L = 1, NEQ
            B(L) = 0.0D0
  101     CONTINUE
          B(I) = 1.0D0
          CALL SOLVE ( A, B, IDIAG, NEQ, .FALSE., .TRUE. )
          DO 110  J = 1, I
            K = K + 1
            AINV(K) = B(J)
  110     CONTINUE
  100   CONTINUE
C
        RETURN
        END
        SUBROUTINE SOLVE (A,B, IDIAG,NEQ, FACT,BACK)
C
C  Compute the U**T * D * U  factorization of the symmetric matrix
C  stored in A, if FACT = TRUE;  and solve A * X = B if BACK = TRUE.
C
C
C  A          Contains the compacted-column form of the upper triangular
C             part of the coefficient matrix.  After factorization, it
```

```
C             contains D and U.
C  B          Right-hand-side vector.  After backsubstitution, it
C             contains the solution.
C
C  IDIAG      Addresses of the diagonal terms in A.
C  NEQ        Number of equations
C
C  FACT       If FACT = TRUE, then factor A;  otherwise do not factor A.
C  BACK       If BACK = TRUE, reduce B and backsubstitute;  otherwise
C             do not solve the equations.
C
C
C
      IMPLICIT REAL*8  (A-H,O-Z)
      LOGICAL FACT,BACK
      DIMENSION A(1),B(1),IDIAG(1)
C
C
C  Factor A, reduce B
      JR = 0
      DO 400  J = 1, NEQ
        JD = IDIAG(J)
        JH = JD - JR
        IS = J - JH + 2
C
        IF (JH .LT. 2)  GOTO 390
C
        IF (FACT) THEN
C
          IF (JH .GT. 2) THEN
C..       Reduce column J rows IS to J-1: do not divide by row diagonal
            K = JR + 2
            ID = IDIAG(IS - 1)
C
            DO 100 I = IS, J-1
              IR = ID
              ID = IDIAG(I)
              IH = MIN (ID-IR-1,I-IS+1)
              IF (IH.GT.0)  A(K) = A(K) - DOT(A(K-IH),A(ID-IH),IH)
              K = K + 1
  100       CONTINUE
          ENDIF
C
C..       Divide by row diagonal, and reduce diagonal term in column J
C
          IR = JR + 1
          K = J - JD
          DO 200  I = IR, JD-1
            ID = IDIAG(K+I)
            IF (A(ID).EQ.0.0) GOTO 200
            D = -A(I)
            A(I) = A(I)/A(ID)
            A(JD) = A(JD) + D*A(I)
  200     CONTINUE
C
        ENDIF
C
C..     Reduce RHS
        IF (BACK)  B(J) = B(J) - DOT(A(JR+1),B(IS-1),JH-1)
C
  390   JR = JD
  400 CONTINUE
C
C
      IF (.NOT.BACK)  RETURN
C
C
C  Divide by diagonal pivots
C
      DO 700 I = 1,NEQ
```

```
          ID = IDIAG(I)
          IF (A(ID).NE.0.0)    B(I) = B(I)/A(ID)
  700 CONTINUE
C
C  Backsubstitute
C
      J = NEQ
      JD = IDIAG(J)
C
  801 D = -B(J)
      J = J - 1
      IF (J.LE.0)  RETURN
C
      JR = IDIAG(J)
      IF (JD-JR.GT.1) THEN
        IS = J - JD + JR + 2
        K = JR - IS + 1
        DO 810  I = IS, J
  810   B(I) = B(I) + A(I+K)*D
      ENDIF
C
      JD = JR
      GOTO 801
C
C
      END
      SUBROUTINE VSMXVT ( A, B, C, JDIAG, NEQ )
C
C.....THIS SUBROUTINE FORMS C = C + A*B WHERE A IS A SYMMETRIC MATRIX
C       STORED IN PROFILE FORM, B, C ARE VECTORS, AND JDIAG LOCATES THE
C       DIAGONALS IN A.
C
      IMPLICIT REAL*8  (A-H,O-Z)
      DIMENSION   A(1), B(1), C(1), JDIAG(1)
C
      JS = 1
      DO 200  J = 1, NEQ
        JD = JDIAG(J)
        IF ( JS. LE. JD )  THEN
          BJ = B(J)
          AB = A(JD)*BJ
          IF ( JS. NE. JD )  THEN
            JB = J - JD
            JE = JD - 1
            DO 100  JJ = JS, JE
              AB = AB + A(JJ)*B(JJ+JB)
              C(JJ+JB) = C(JJ+JB) + A(JJ)*BJ
  100       CONTINUE
          END IF
          C(J) = C(J) + AB
        END IF
        JS = JD + 1
  200 CONTINUE
C
      RETURN
      END
      SUBROUTINE DCLR (A,NA)
C
      INTEGER  NA, I
      REAL*8   A(1)
C
      DO 100  I = 1, NA
        A(I) = 0.0D0
  100 CONTINUE
C
      RETURN
      END
//*
// EXEC LINKGOV,REGION=5000K
//*
```

```
//SYSLIB   DD DSN=SYS2.IMSLD,DISP=SHR
//*        DD DSN=SYS2.IMSLS,DISP=SHR
//*
//GO.FT05F001 DD DSN=ZMAYA.ALTIM.DATA(BE180),DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** *
//*                                                        *
//*         BLACK SEA DATA AFTER SORT                      *
//*                                                        *
//* **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT06F001 DD SYSOUT=*
//GO.FT08F001 DD DSN=ZMAYA.BLACK4.SORT,DISP=SHR
//GO.FT09F001 DD DSN=ZMAYA.BLACKC4H.GRID,DISP=SHR
//GO.FT10F001 DD DSN=ZMAYA.BLACKC4H.COVM,DISP=SHR
//GO.FT11F001 DD DSN=ZMAYA.BLACK.PCOV,DISP=SHR
// EXEC NOTIFYTS
```

A listing of PROGRAM EMPCOV

```
//ZMAYAB36 JOB (G0109,360,2),AYAU,TIME=(00,30),CLASS=F,MSGCLASS=X
// EXEC FORTVC
//SYSIN DD *
C
C
      PARAMETER ( XMIN = 26.5 )
      PARAMETER ( XMAX = 42.5 )
      PARAMETER ( YMIN = 40.0 )
      PARAMETER ( YMAX = 48.0 )
      PARAMETER ( DXY = 0.25 )
      PARAMETER ( M = (XMAX-XMIN)/DXY + 1 )
      PARAMETER ( N = (YMAX-YMIN)/DXY + 1 )
      PARAMETER ( MN = N )
C
      IMPLICIT REAL*8 (A-Z)
      DIMENSION  G(M,N), DG(M,N), H(M,N), DH(M,N)
      DIMENSION  D(0:MN)
      DIMENSION  CHH(0:MN),CHHNS(0:MN), CHHEW(0:MN)
      DIMENSION  CHG(0:MN),CHGNS(0:MN), CHGEW(0:MN)
      DIMENSION  CGG(0:MN),CGGNS(0:MN), CGGEW(0:MN)
      INTEGER    INP, OUP, IMH, INH, IMG, ING, OUF
C
      DATA   INP / 5 /, OUP / 6 /, IMH / 7 /, INH / 8 /,
     *       IMG / 9 /, ING / 10 /, OUF / 11 /
C
C
C.....READ IN THE MODEL GEOIDAL HEIGHT AND GRAVITY ANOMALY
C
      DO 100  J = N, 1, -1
        READ ( IMH, * )  RL
        READ ( IMH, 501 )  (H(I,J),I=1,M)
  100 CONTINUE
C
      DO 101  J = N, 1, -1
        READ ( IMG, * )  RL
        READ ( IMG, 501 )  (G(I,J),I=1,M)
  101 CONTINUE
C
C.....READ IN THE GEOIDAL HEIGHT AND GRAVITY ANOMALY DATA
C
      DO 110  I = 1, M
        READ ( INH, 502 )  (DH(I,J),J=1,N)
  110 CONTINUE
C
      DO 111  I = 1, M
        READ ( ING, 502 )  (DG(I,J),J=1,N)
  111 CONTINUE
C
C.....REMOVE THE MODEL VALUE FROM THE DATA
C
      DO 120  I = 1, M
        DO 121  J = 1, N
          IF ( DH(I,J) .NE. -1000.0 )  DH(I,J) = DH(I,J) - H(I,J)
          IF ( DG(I,J) .NE. -1000.0 )  DG(I,J) = DG(I,J) - G(I,J)
  121   CONTINUE
  120 CONTINUE
C
C.....PERFORM THE CONVOLUTION
C
      DO 200  K = 0, MN-1
C
        SHH = 0.
        SHG = 0.
        SGG = 0.
        CHHNS(K) = 0.
        CHGNS(K) = 0.
        CGGNS(K) = 0.
```

```
      DO 210  J = 1, N
        DO 211  I = K+1, M
          IF ( DH(I-K,J).NE.-1000.0 .AND. DH(I,J).NE.-1000.0 )  THEN
            SHH = SHH + DH(I-K,J)*DH(I,J)
          END IF
          IF ( DH(I-K,J).NE.-1000.0 .AND. DG(I,J).NE.-1000.0 )  THEN
            SHG = SHG + DH(I-K,J)*DG(I,J)
          END IF
          IF ( DG(I-K,J).NE.-1000.0 .AND. DG(I,J).NE.-1000.0 )  THEN
            SGG = SGG + DG(I-K,J)*DG(I,J)
          END IF
  211   CONTINUE
  210 CONTINUE
      C1 = ((M-K)*N)
      CHHNS(K) = SHH / C1
      CHGNS(K) = SHG / C1
      CGGNS(K) = SGG / C1
C
      SHH = 0.
      SHG = 0.
      SGG = 0.
      CHHEW(K) = 0.
      CHGEW(K) = 0.
      CGGEW(K) = 0.
      DO 220  I = 1, M
        DO 221  J = K+1, N
          IF ( DH(I,J-K).NE.-1000.0 .AND. DH(I,J).NE.-1000.0 )  THEN
            SHH = SHH + DH(I,J-K)*DH(I,J)
          END IF
          IF ( DH(I,J-K).NE.-1000.0 .AND. DG(I,J).NE.-1000.0 )  THEN
            SHG = SHG + DH(I,J-K)*DG(I,J)
          END IF
          IF ( DG(I,J-K).NE.-1000.0 .AND. DG(I,J).NE.-1000.0 )  THEN
            SGG = SGG + DG(I,J-K)*DG(I,J)
          END IF
  221   CONTINUE
  220 CONTINUE
      C2 = (M*(N-K))
      CHHEW(K) = SHH / C2
      CHGEW(K) = SHG / C2
      CGGEW(K) = SGG / C2
C
      D(K) = K * DXY
      CHH(K) = ( C1*CHHNS(K) + C2*CHHEW(K) ) / ( C1 + C2 )
      CHG(K) = ( C1*CHGNS(K) + C2*CHGEW(K) ) / ( C1 + C2 )
      CGG(K) = ( C1*CGGNS(K) + C2*CGGEW(K) ) / ( C1 + C2 )
C
  200 CONTINUE
C
      WRITE ( OUP, 601 )
      WRITE ( OUP, 602 )  ( D(I), CHH(I), CHG(I), CGG(I), I = 0, MN-1 )
      WRITE ( OUF, 603 )  ( D(I), CHH(I), CHG(I), CGG(I), I = 0, MN-1 )
C
C
  501 FORMAT((6(2X,F10.3)))
  502 FORMAT((3(14X,F12.4)))
  601 FORMAT(1H1,///,T5,'SPHERICAL DISTANCE',T30,'C(NN) IN M**2',
     *               T48,'C(NG) IN M*MGAL',T67,'C(GG) IN MGAL**2'/,
     *               T5,'------------------',T30,'-------------',
     *               T48,'---------------',T67,'----------------',/)
  602 FORMAT((1H ,T15,F5.2,T30,F12.4,T50,F12.4,T70,F12.4))
  603 FORMAT((1X,D10.4,3D20.13))
C
      STOP
      END
//*
// EXEC LINKGOV,REGION=5000K
//*
//GO.FT06F001 DD SYSOUT=*
//GO.FT07F001 DD DSN=ZMAYA.ALTIM.DATA(BRH36),DISP=SHR
```

```
//GO.FT08F001 DD DSN=ZMAYA.ALTIM.DATA(RDH300),DISP=SHR
//GO.FT09F001 DD DSN=ZMAYA.ALTIM.DATA(BRG36),DISP=SHR
//GO.FT10F001 DD DSN=ZMAYA.ALTIM.DATA(RDG300),DISP=SHR
//GO.FT11F001 DD DSN=ZMAYA.ALTIM.DATA(BEI036),DISP=SHR
//*
// EXEC NOTIFYTS
```

A listing of PROGRAM GRAVEN

```
//ZMAYAB0R JOB (G0109,360,2),AYAU,TIME=(04,00),CLASS=O,MSGCLASS=X
/*JOBPARM LINES=60
// EXEC FORTVC
//SYSIN DD *
C
      PARAMETER ( MODEL = 180 )
      PARAMETER ( ANOISE = 0.25 )
      PARAMETER ( ITM = 0 )
      PARAMETER ( CAP = 1.00 )
      PARAMETER ( IPRT = 0 )
      PARAMETER ( CNVRGE = 1.0E-4 )
      PARAMETER ( SCX = 1.00 )
      PARAMETER ( KUTOFF = 3 )
      PARAMETER ( MTR = 600 )
      PARAMETER ( XMIN = 26.5 )
      PARAMETER ( XMAX = 42.5 )
      PARAMETER ( YMIN = 40.0 )
      PARAMETER ( YMAX = 48.0 )
      PARAMETER ( XG1 = 26.5 )
      PARAMETER ( XG2 = 42.5 )
      PARAMETER ( YG1 = 40.0 )
      PARAMETER ( YG2 = 48.0 )
      PARAMETER ( DX = 0.25 )
      PARAMETER ( DY = 0.25 )
      PARAMETER ( NXI = (XG2-XG1)/DX + 1 )
      PARAMETER ( NYI = (YG2-YG1)/DY + 1 )
      PARAMETER ( M1 = (XMIN-XG1)/DX + 1 )
      PARAMETER ( M2 = (XMAX-XG1)/DX + 1 )
      PARAMETER ( N1 = (YMIN-YG1)/DY + 1 )
      PARAMETER ( N2 = (YMAX-YG1)/DY + 1 )
      PARAMETER ( LCT = NYI )
      PARAMETER ( INC = 4. * CAP + 1 )
      PARAMETER ( CAP2 = CAP*CAP )
      PARAMETER ( LEN = (MTR*(MTR+1))/2 )
      PARAMETER ( LCTF = (2*LCT-1) )
      PARAMETER ( NFFT = (6*LCTF/2)+150 )
C
      REAL*4      GRIDX(NXI), GRIDY(NYI)
      REAL*4      GUI(NXI,NYI), HVARI(NXI,NYI)
      REAL*4      GU(NXI,NYI), HVAR(NXI,NYI)
      REAL*4      GA(NXI,NYI), GVAR(NXI,NYI)
      REAL*4      DIFF(NXI,NYI)
      REAL*4      GRA300(NXI,NYI), GUN300(NXI,NYI)
      REAL*4      GRA180(NXI,NYI), GUN180(NXI,NYI)
      REAL*4      GRA036(NXI,NYI), GUN036(NXI,NYI)
      REAL*4      GRAN(NXI,NYI), GUND(NXI,NYI)
      REAL*4      WKCG(NXI,NYI), WKCH(NXI,NYI)
      REAL*4      CHHNS(0:NYI-1), CHHEW(0:NYI-1)
      REAL*4      CHGNS(0:NYI-1), CHGEW(0:NYI-1)
      REAL*4      CGGNS(0:NYI-1), CGGEW(0:NYI-1)
      REAL*8      RGRIDX(NXI), RGRIDY(NYI)
      REAL*8      RMSH(0:ITM), RMSG(0:ITM)
      REAL*8      RMSH1(0:ITM), RMSG1(0:ITM)
      REAL*8      RMSH2(0:ITM), RMSG2(0:ITM)
      REAL*8      RMSH3(0:ITM), RMSG3(0:ITM)
      REAL*8      SY(MTR), CY(MTR), SE(MTR)
      REAL*8      EX(MTR), GY(MTR), GH(MTR), SGY(MTR), CGY(MTR)
      REAL*8      CM(LEN), CMI(LEN), CMS(LEN)
      REAL*8      CP(MTR), SRC(MTR), AP(MTR)
      REAL*8      SD(LCT), CTV(LCT,3)
      REAL*8      CVUU(LCT), SCUU(LCT-1,3)
      REAL*8      CVUG(LCT), SCUG(LCT-1,3)
      REAL*8      CVGG(LCT), SCGG(LCT-1,3)
      REAL*8      WKF(NFFT), A(LCTF)
      REAL*8      PCVUU(LCTF), PCVUG(LCTF), PCVGG(LCTF)
      REAL*8      DTR, ELON, GLAT, X1, Y1, SY1, CY1, PVAL, VAR
      COMPLEX*16  X(LCTF)
```

```
      INTEGER    IDIAG(MTR), MASK(NXI,NYI), IWKF(NFFT)
      INTEGER    INP, OUP, INH, INV, INU1, ING1, INU2, ING2, INM
      INTEGER    OUG, OVG, OUH, OVH, OUD, OUB
C
      DATA   INP / 5 /, OUP / 6 /, INH / 8 /, INV / 9 /
      DATA   INU1 / 10 /, ING1 / 11 /, INU2 / 12 /, ING2 / 13 /
      DATA   INU3 / 14 /, ING3 / 15 /
      DATA   INM / 16 /
      DATA   OUG / 17 /, OVG / 18 /, OUH / 19 /, OVH / 20 /, OUD / 21 /
      DATA   OUB / 22 /
      DATA   NIER / 0 /
      DATA   DTR / 3.490658504D-2 /
      DATA   GMING / 1000.0 /, GMAXG / -999.0 /
      DATA   VMING / 1000.0 /, VMAXG / -999.0 /
      DATA   GMINH / 1000.0 /, GMAXH / -999.0 /
      DATA   VMINH / 1000.0 /, VMAXH / -999.0 /
C
C.....READ IN THE BLACK SEA MASK
C
      READ ( INM, 505 )  ((MASK(I,J),I=M1,M2),J=N2,N1,-1)
      WRITE ( OUP, 612 )
      WRITE ( OUP, 505 )  ((MASK(I,J),I=M1,M2),J=N2,N1,-1)
C
C.....GENERATE A GRID NET OF SINE AND COSINE
C
      GRIDX(1) = XG1
      RGRIDX(1) = DBLE ( GRIDX(1) ) * DTR
      DO 100  I = 2, NXI
        GRIDX(I) = GRIDX(I-1) + DX
        RGRIDX(I) = DBLE ( GRIDX(I) ) * DTR
  100 CONTINUE
      GRIDY(1) = YG1
      RGRIDY(1) = DBLE ( GRIDY(1) ) * DTR
      SY(1) = DSIN(RGRIDY(1))
      CY(1) = DCOS(RGRIDY(1))
      DO 101  I = 2, NYI
        GRIDY(I) = GRIDY(I-1) + DY
        RGRIDY(I) = DBLE ( GRIDY(I) ) * DTR
        SY(I) = DSIN(RGRIDY(I))
        CY(I) = DCOS(RGRIDY(I))
  101 CONTINUE
C
C.....READ IN THE MODEL GEOIDAL HEIGHT AND GRAVITY ANOMALY
C.....  THE OSU 180X180, 36X36 AND 300X300 MODELS
C
      DO 102  J = NYI, 1, -1
        READ ( INU1, * )  RL
        READ ( INU1, 503 )  (GUN180(I,J),I=1,NXI)
  102 CONTINUE
C
      DO 103  J = NYI, 1, -1
        READ ( ING1, * )  RL
        READ ( ING1, 503 )  (GRA180(I,J),I=1,NXI)
  103 CONTINUE
C
      DO 104  J = NYI, 1, -1
        READ ( INU2, * )  RL
        READ ( INU2, 503 )  (GUN036(I,J),I=1,NXI)
  104 CONTINUE
C
      DO 105  J = NYI, 1, -1
        READ ( ING2, * )  RL
        READ ( ING2, 503 )  (GRA036(I,J),I=1,NXI)
  105 CONTINUE
C
      DO 106  J = NYI, 1, -1
        READ ( INU3, * )  RL
        READ ( INU3, 503 )  (GUN300(I,J),I=1,NXI)
  106 CONTINUE
C
```

```
      DO 107  J = NYI, 1, -1
        READ ( ING3, * )  RL
        READ ( ING3, 503 )  (GRA300(I,J),I=1,NXI)
  107 CONTINUE
C
C.....READ IN THE GRIDDED GEOID UNDULATION DATA AND VARIANCES
C
      DO 108  I = M1, M2
        READ ( INH, 504 )  (GRIDX(I),GRIDY(J),GUI(I,J),J=N1,N2)
  108 CONTINUE
C
      DO 109  I = M1, M2
        READ ( INV, 504 )  (GRIDX(I),GRIDY(J),HVARI(I,J),J=N1,N2)
  109 CONTINUE
C
      IF ( IPRT .NE. 0 )  THEN
        WRITE ( OUP, 608 )
        WRITE ( OUP, 609 )  ((GRIDX(I),GRIDY(J),GRA036(I,J),GUN036(I,J),
     *                        GRA180(I,J),GUN180(I,J),GRA300(I,J),
     *                        GUN300(I,J),GUI(I,J),HVARI(I,J),
     *                        J=1,NYI),I=1,NXI)
      END IF
C
C.....SELECT THE APPROPRIATE OSU MODEL
C
      IF ( MODEL .EQ. 180 )  THEN
C
        DO 110  I = M1, M2
          DO 111  J = N1, N2
            GRAN(I,J) = GRA180(I,J)
            GUND(I,J) = GUN180(I,J)
  111     CONTINUE
  110   CONTINUE
C
      ELSE IF ( MODEL .EQ. 36 )  THEN
C
        DO 112  I = M1, M2
          DO 113  J = N1, N2
            GRAN(I,J) = GRA036(I,J)
            GUND(I,J) = GUN036(I,J)
  113     CONTINUE
  112   CONTINUE
C
      ELSE IF ( MODEL .EQ. 300 )  THEN
C
        DO 114  I = M1, M2
          DO 115  J = N1, N2
            GRAN(I,J) = GRA300(I,J)
            GUND(I,J) = GUN300(I,J)
  115     CONTINUE
  114   CONTINUE
C
      ELSE
C
        WRITE ( OUP, 999 )  MODEL
        STOP
C
      END IF
C
C.....READ IN THE COVARIANCE TABLE
C
        READ ( INP, 501 )  ( SD(I), (CTV(I,J),J=1,3), I=1,LCT)
C
C.....SCALE THE CORELATION LENGTH OF THE COVARIANCE FUNCTION
C
        DO 149  I = 1, LCT
          SD(I) = SD(I) * DBLE(SCX)
  149   CONTINUE
C
C.....CONSTRUCT THE COVARIANCE MATRIX DIAGONAL ADDRESS
```

```
C
      DO 116  I = 1, MTR
        IDIAG(I) = (I*(I+1))/2
  116 CONTINUE
C
C
C.....START AN ITERATION PROCESS TO REFINE THE GRAVITY ANOMALY
C
      DO 900  IJK = 0, ITM
C
        DO 150  I = 1, LCT
          CVUU(I) = CTV(I,1)
          CVUG(I) = CTV(I,2)
          CVGG(I) = CTV(I,3)
  150   CONTINUE
C
C.....DETERMINE THE POWER SPECTRA OF THE COVARIANCE FUNCTIONS
C
        CALL POWSPC ( A, CVUU, X, PCVUU, LCT, LCTF, NFFT, IWKF, WKF )
        CALL POWSPC ( A, CVUG, X, PCVUG, LCT, LCTF, NFFT, IWKF, WKF )
        CALL POWSPC ( A, CVGG, X, PCVGG, LCT, LCTF, NFFT, IWKF, WKF )
C
C.....DETERMINE A CUBIC SPLINE COEFFICIENT MATRIX SC
C
        CALL ICSCCU ( SD, CVUU, LCT, SCUU, LCT-1, IER )
        CALL ICSCCU ( SD, CVUG, LCT, SCUG, LCT-1, IER )
        CALL ICSCCU ( SD, CVGG, LCT, SCGG, LCT-1, IER )
C
C.....WRITE OUT THE VALUES OF THE COVARIANCE FUNCTIONS AND THE
C     CORRESPONDING POWER SPECTRA
C
        WRITE ( OUB, 631 )  IJK
        WRITE ( OUB; 501 )  (SD(I),CTV(I,1),CTV(I,2),CTV(I,3),I=1,LCT)
        WRITE ( OUB, 632 )  IJK
        WRITE ( OUB, 501 )  (SD(I),PCVUU(I),PCVUG(I),PCVGG(I),I=1,LCT)
        WRITE ( OUP, 601 )  IJK
        WRITE ( OUP, 602 )  (SD(I),CTV(I,1),CTV(I,2),CTV(I,3),I=1,LCT)
        WRITE ( OUP, 634 )  IJK
        WRITE ( OUP, 602 )  (SD(I),PCVUU(I),PCVUG(I),PCVGG(I),I=1,LCT)
C
C.....TRANSFORM THE GEOIDAL HEIGHT TO GRAVITY ANOMALY
C
        WRITE ( OUP, 600 )
        WRITE ( OUP, * )  '     GEOIDAL HEIGHT TO GRAVITY ANOMALY: ', IJK
        WRITE ( OUP, * )  '
C
        RMS = 0.
        RMS1 = 0.
        RMS2 = 0.
        RMS3 = 0.
        JCG = 0
        DO 200  I = M1, M2
          DO 201  J = N1, N2
C
            GA(I,J) = -1000.0
            GVAR(I,J) = -1000.0
C
            IF ( MASK(I,J) .NE. 0 .AND. GRAN(I,J) .NE. -1000.0 .AND.
     *        GUI(I,J) .NE. -1000.0 .AND. GUND(I,J) .NE. -1000.0 )  THEN
C
              I1 = I - INC
              I2 = I + INC
              J1 = J - INC
              J2 = J + INC
              IF ( I1. LT. 1 )  I1 = 1
              IF ( I2. GT. NXI )  I2 = NXI
              IF ( J1. LT. 1 )  J1 = 1
              IF ( J2. GT. NYI )  J2 = NYI
C
              IREC = 0
```


C.2

```
C
                DO 210  M = I1, I2
                  DO 211  N = J1, J2
C
                    IF (GUI(M,N).NE.-1000..AND.GUND(M,N).NE.-1000.)  THEN
                      RS = (GRIDX(I)-GRIDX(M))**2 + (GRIDY(J)-GRIDY(N))**2
                      IF ( RS. LT. CAP2 )  THEN
                        IREC = IREC + 1
                        EX(IREC) = RGRIDX(M)
                        GY(IREC) = RGRIDY(N)
                        GH(IREC) = DBLE ( GUI(M,N) - GUND(M,N) )
                        SE(IREC) = DBLE ( 0.01*HVARI(M,N) + ANOISE )
                      END IF
                    END IF
C
  211             CONTINUE
  210           CONTINUE
C
                IF ( IREC. GT. KUTOFF ) THEN
                  X1 = RGRIDX(I)
                  Y1 = RGRIDY(J)
                  SY1 = SY(J)
                  CY1 = CY(J)
                  JREC = (IREC*(IREC+1))/2
                  JCG = JCG + 1
                  CALL PREDIC ( 1, PVAL,VAR, EX,GY,GH,SGY,CGY,IREC,
     *                            SE, X1,Y1,SY1,CY1,
     *                            CM,CMS,CMI,JREC, IDIAG,CP,SRC,AP,
     *                            SD,CTV,CVUU,SCUU,CVUG,SCUG,CVGG,SCGG,
     *                            LCT, DTR, IER, NIER )
                  IF ( IER. EQ. 0 )  THEN
                    GA(I,J) = PVAL + GRAN(I,J)
                    GVAR(I,J) = VAR
                    RMS = RMS + PVAL*PVAL
                    D1 = GA(I,J) - GRA180(I,J)
                    D2 = GA(I,J) - GRA036(I,J)
                    D3 = GA(I,J) - GRA300(I,J)
                    RMS1 = RMS1 + D1*D1
                    RMS2 = RMS2 + D2*D2
                    RMS3 = RMS3 + D3*D3
                  ELSE
                    JCG = JCG - 1
                    IF ( IPRT .NE. 0 )  WRITE ( OUP, * )  'IER =',IER,
     *                    '      PVAL =',PVAL,'      VAR =',VAR
                  END IF
                  IF ( IPRT .NE. 0 )  WRITE ( OUP, 603 )  I,J,GRIDX(I),
     *                  GRIDY(J),GA(I,J), GVAR(I,J), PVAL, IREC,JCG
                END IF
C
              END IF
C
  201       CONTINUE
  200     CONTINUE
C
          RMSG(IJK) = SQRT ( RMS / FLOAT(JCG) )
          RMSG1(IJK) = SQRT ( RMS1 / FLOAT(JCG) )
          RMSG2(IJK) = SQRT ( RMS2 / FLOAT(JCG) )
          RMSG3(IJK) = SQRT ( RMS3 / FLOAT(JCG) )
C
          WRITE ( OUP, 614 )  RMSG(IJK),RMSG3(IJK),RMSG1(IJK),RMSG2(IJK)
C
C.....TRANSFORM BACK FROM GRAVITY ANOMALY TO GEOID HEIGHT
C
          WRITE ( OUP, 600 )
          WRITE ( OUP, * )  '    GRAVITY ANOMALY BACK TO GEOIDAL HEIGHT:',
     *                      IJK
          WRITE ( OUP, * )  '     '
C
          RMS = 0.
          RMS1 = 0.
```

```
      RMS2 = 0.
      RMS3 = 0.
      JCH = 0
      DO 250  I = M1, M2
        DO 251  J = N1, N2
C
          GU(I,J) = -1000.0
          HVAR(I,J) = -1000.0
C
          IF ( MASK(I,J) .NE. 0 .AND. GUND(I,J) .NE. -1000.0 .AND.
     *       GA(I,J) .NE. -1000.0 .AND. GRAN(I,J) .NE. -1000.0 )  THEN
C
            I1 = I - INC
            I2 = I + INC
            J1 = J - INC
            J2 = J + INC
            IF ( I1. LT. 1 )  I1 = 1
            IF ( I2. GT. NXI )  I2 = NXI
            IF ( J1. LT. 1 )  J1 = 1
            IF ( J2. GT. NYI )  J2 = NYI
C
            IREC = 0
C
            DO 260  M = I1, I2
              DO 261  N = J1, J2
C
                IF (GA(M,N).NE.-1000..AND.GRAN(M,N).NE.-1000.)  THEN
                  RS = (GRIDX(I)-GRIDX(M))**2 + (GRIDY(J)-GRIDY(N))**2
                  IF ( RS. LT. CAP2 )  THEN
                    IREC = IREC + 1
                    EX(IREC) = RGRIDX(M)
                    GY(IREC) = RGRIDY(N)
                    GH(IREC) = DBLE ( GA(M,N) - GRAN(M,N) )
                    SE(IREC) = DBLE ( GVAR(M,N) )
                  END IF
                END IF
C
  261           CONTINUE
  260         CONTINUE
C
            IF ( IREC. GT. KUTOFF ) THEN
              X1 = RGRIDX(I)
              Y1 = RGRIDY(J)
              SY1 = SY(J)
              CY1 = CY(J)
              JREC = (IREC*(IREC+1))/2
              JCH = JCH + 1
              CALL PREDIC ( 2, PVAL,VAR, EX,GY,GH,SGY,CGY,IREC,
     *                        SE, X1,Y1,SY1,CY1,
     *                        CM,CMS,CMI,JREC, IDIAG,CP,SRC,AP,
     *                        SD,CTV,CVUU,SCUU,CVUG,SCUG,CVGG,SCGG,
     *                        LCT, DTR, IER, NIER )
              IF ( IER. EQ. 0 )  THEN
                GU(I,J) = PVAL + GUND(I,J)
                HVAR(I,J) = 100.*VAR
                RMS = RMS + PVAL*PVAL
                D1 = GU(I,J) - GUN180(I,J)
                D2 = GU(I,J) - GUN036(I,J)
                D3 = GU(I,J) - GUN300(I,J)
                RMS1 = RMS1 + D1*D1
                RMS2 = RMS2 + D2*D2
                RMS3 = RMS3 + D3*D3
              ELSE
                JCH = JCH - 1
                IF ( IPRT .NE. 0 )  WRITE ( OUP, * )  'IER =',IER,
     *                '      PVAL =',PVAL,'      VAR =',VAR
              END IF
              IF ( IPRT .NE. 0 )  WRITE ( OUP, 617 )  I,J,GRIDX(I),
     *                GRIDY(J), GU(I,J), HVAR(I,J), PVAL, IREC,JCH
            END IF
```

```
C
            END IF
C
  251      CONTINUE
  250    CONTINUE
C
         RMSH(IJK) = SQRT ( RMS / FLOAT(JCH) )
         RMSH1(IJK) = SQRT ( RMS1 / FLOAT(JCH) )
         RMSH2(IJK) = SQRT ( RMS2 / FLOAT(JCH) )
         RMSH3(IJK) = SQRT ( RMS3 / FLOAT(JCH) )
C
         WRITE ( OUP, 614 )  RMSH(IJK),RMSH3(IJK),RMSH1(IJK),RMSH2(IJK)
C
C.....CHECK CONVERGENCE CRITERION
C
         IF ( RMSG(IJK) .LT. CNVRGE )  GO TO 777
C
C.....REDETERMINE THE EMPIRICAL COVARIANCE FUNCTION
C
          CALL EMPCVF ( SD,CTV, NXI,NYI,DX,NYI, GRAN,GA, GUND,GUI,
     *                  WKCG,WKCH, CHHNS,CHHEW, CHGNS,CHGEW, CGGNS,CGGEW )
C
C.....USE THE RESULTANT GEOIDAL HEIGHT AND GRAVITY ANOMALY AS A
C       WORKING MODEL FOR THE NEXT ITERATION STEP
C
         DO 270   I = 1, NXI
           DO 271  J = 1, NYI
             GUND(I,J) = GU(I,J)
  271      CONTINUE
  270    CONTINUE
C
         DO 280   I = 1, NXI
           DO 281  J = 1, NYI
             GRAN(I,J) = GA(I,J)
  281      CONTINUE
  280    CONTINUE
C
  900 CONTINUE
C
C
C.....WRITE OUT THE SUMMARY OF THE ITERATION.
C
  777 IF ( IJK .GT. ITM )  IJK = ITM
      WRITE ( OUP, 618 )  MODEL
      WRITE ( OUP, 621 )  ( L, RMSG(L), RMSH(L), RMSG3(L), RMSH3(L),
     *                      RMSG1(L), RMSH1(L), RMSG2(L), RMSH2(L),
     *                                          L = 0, IJK )
C
C.....DETERMINE THE RANGE OF THE PREDICTED VALUES AND THEIR VARIANCE
C
       DO 300  I = M1, M2
         DO 301  J = N1, N2
           IF ( GA(I,J). NE. -1000.0 )  THEN
             GMING = AMIN1 ( GMING, GA(I,J) )
             GMAXG = AMAX1 ( GMAXG, GA(I,J) )
           END IF
           IF ( GVAR(I,J). NE. -1000.0 )  THEN
             VMING = AMIN1 ( VMING, GVAR(I,J) )
             VMAXG = AMAX1 ( VMAXG, GVAR(I,J) )
           END IF
           IF ( GU(I,J). NE. -1000.0 )  THEN
             GMINH = AMIN1 ( GMINH, GU(I,J) )
             GMAXH = AMAX1 ( GMAXH, GU(I,J) )
           END IF
           IF ( HVAR(I,J). NE. -1000.0 )  THEN
             VMINH = AMIN1 ( VMINH, HVAR(I,J) )
             VMAXH = AMAX1 ( VMAXH, HVAR(I,J) )
           END IF
  301    CONTINUE
  300  CONTINUE
```

```
C
C.....PRINT OUT THE GRIDDED VALUES AND THE VARIANCE
C
      NXY = (M2-M1+1)*(N2-N1+1)
      WRITE ( OUP, 604 )  NIER, JCG, NXY, GMING, GMAXG
      DO 400  I = M1, M2
        WRITE ( OUP, 605 )  (GRIDX(I),GRIDY(J),GA(I,J),J=N1,N2)
        WRITE ( OUG, 606 )  (GRIDX(I),GRIDY(J),GA(I,J),J=N1,N2)
  400 CONTINUE
C
      WRITE ( OUP, 615 )  NIER, JCH, NXY, GMINH, GMAXH
      DO 401  I = M1, M2
        WRITE ( OUP, 605 )  (GRIDX(I),GRIDY(J),GU(I,J),J=N1,N2)
C       WRITE ( OUH, 606 )  (GRIDX(I),GRIDY(J),GU(I,J),J=N1,N2)
  401 CONTINUE
C
      WRITE ( OUP, 607 )  VMING, VMAXG
      DO 410  I = M1, M2
        WRITE ( OUP, 605 )  (GRIDX(I),GRIDY(J),GVAR(I,J),J=N1,N2)
        WRITE ( OVG, 606 )  (GRIDX(I),GRIDY(J),GVAR(I,J),J=N1,N2)
  410 CONTINUE
C
      WRITE ( OUP, 616 )  VMINH, VMAXH
      DO 411  I = M1, M2
        WRITE ( OUP, 605 )  (GRIDX(I),GRIDY(J),HVAR(I,J),J=N1,N2)
C       WRITE ( OVH, 606 )  (GRIDX(I),GRIDY(J),HVAR(I,J),J=N1,N2)
  411 CONTINUE
C
C.....DETERMINE THE DIFFERENCE BETWEEN THE PREDICTED GRAVITY ANOMALIES
C       AND THE MODEL VALUES USED
C
      IF ( MODEL .EQ. 180 )  THEN
C
        DO 420  I = 1, NXI
          DO 421  J = 1, NYI
           DIFF(I,J) = -1000.0
           IF ( GA(I,J) .NE. -1000.0 )  THEN
             DIFF(I,J) = GA(I,J) - GRA180(I,J)
           END IF
  421     CONTINUE
  420   CONTINUE
C
      ELSE IF ( M .EQ. 36 )  THEN
C
        DO 430  I = 1, NXI
          DO 431  J = 1, NYI
            DIFF(I,J) = -1000.0
            IF ( GA(I,J) .NE. -1000.0 )  THEN
              DIFF(I,J) = GA(I,J) - GRA036(I,J)
            END IF
  431     CONTINUE
  430   CONTINUE
C
      ELSE IF ( M .EQ. 300 )  THEN
C
        DO 440  I = 1, NXI
          DO 441  J = 1, NYI
            DIFF(I,J) = -1000.0
            IF ( GA(I,J) .NE. -1000.0 )  THEN
              DIFF(I,J) = GA(I,J) - GRA300(I,J)
            END IF
  441     CONTINUE
  440   CONTINUE
C
      END IF
C
      WRITE ( OUP, 622 )
      DO 450  I = M1, M2
        WRITE ( OUP, 605 )  (GRIDX(I),GRIDY(J),DIFF(I,J),J=N1,N2)
        WRITE ( OUD, 606 )  (GRIDX(I),GRIDY(J),DIFF(I,J),J=N1,N2)
```

```
  450 CONTINUE
C
C
  501 FORMAT(1X,D10.4,3D20.13)
C 502 FORMAT((4D20.13))
  503 FORMAT((6(2X,F10.3)))
  504 FORMAT((3(2F7.2,F12.4)))
  505 FORMAT((3X,65I1))
  600 FORMAT(1H1)
  601 FORMAT(1H1,T5,'VALUES OF THE COVARIANCE TABLE USED',
     *            ' IN ITERATION: ',I3,/1H0,/)
  602 FORMAT((1H ,5X,D10.4,3(5X,D20.13)))
  603 FORMAT(1H0,T2,'GRID(',I2,',',I2,')',2X,'ELON = ',F6.2,1X,
     *       ',GLAT = ',F6.2,5X,'GRAVITY ANOM = ',F10.4,' +- ',F6.2,
     *       ',  DELTA = ',F6.2,5X,'REC #: ',I3,2X,I8)
  604 FORMAT(1H1,T5,I4,' GRID POINTS HAVING NEGATIVE VARIANCE',/,1H0,/,
     *       T5,'THERE ARE ',I4,' POINTS GRIDDED OUT OF A NET ',
     *       ' OF ',I5,' GRID POINTS.',/,1H0,/,T8,'THE RANGE OF THE',
     *       ' GRAVITY ANOMALY IN MGAL IS:',2F12.4,/,1H0)
  605 FORMAT((1H ,5(2F7.2,F12.4)))
  606 FORMAT((3(2F7.2,F12.4)))
  607 FORMAT(1H1,T5,'THE RANGE OF THE VARIANCE IN MGAL IS:',2F12.4,/1H0)
  608 FORMAT(1H1,T10,'MODEL GRAVITY ANOMALY AND GEOID UNDULATION:',/1H0)
  609 FORMAT((1H ,T5,2F10.2,6F12.3,4X,2F14.4))
  610 FORMAT(1H1,T10,'INPUT GEOID UNDULATION:',/,1H0)
  611 FORMAT(1H1,T10,'INPUT GEOID STANDARD ERROR:',/,1H0)
  612 FORMAT(1H1,T10,'THE BLACK SEA MASK:',/,1H0)
  613 FORMAT((1X,5(2F7.2,F12.4)))
  614 FORMAT(1H0,//,T8,'THE RELATIVE RMS DIFFERENCES IS : ',F10.4,/,
     *           T8,'THE RMS RESPECT TO MODEL 300 IS : ',F10.4,/,
     *           T8,'THE RMS RESPECT TO MODEL 180 IS : ',F10.4,/,
     *           T8,'THE RMS RESPECT TO MODEL  36 IS : ',F10.4)
  615 FORMAT(1H1,T5,I4,' GRID POINTS HAVING NEGATIVE VARIANCE',/,1H0,/,
     *       T5,'THERE ARE ',I4,' POINTS GRIDDED OUT OF A NET ',
     *       ' OF ',I5,' GRID POINTS.',/,1H0,/,T8,'THE RANGE OF THE',
     *       ' GEOIDAL HEIGHT IN METER IS:',2F12.4,/,1H0)
  616 FORMAT(1H1,T5,'THE RANGE OF THE VARIANCE IN CM IS:',2F12.4,/1H0)
  617 FORMAT(1H0,T2,'GRID(',I2,',',I2,')',2X,'ELON = ',F6.2,1X,
     *       ',GLAT = ',F6.2,5X,'GEOID HEIGHT = ',F10.4,' +- ',F6.2,
     *       ',  DELTA = ',F6.2,5X,'REC #: ',I3,2X,I8)
  618 FORMAT(1H1,///,T30,'OSU MODEL USED: ',I3,////,
     *           T10,'SUMMARY OF THE RMS OF THE ITERATIONS:',///,
     *           T10, '   RELATIVE    ', T26,'  RELATIVE  ',
     *           T40, '   300 X 300   ', T56,' 300 X 300  ',
     *           T70, '   180 X 180   ', T86,' 180 X 180  ',
     *           T100,'    36 X 36    ',T116,'  36 X 36   '/
     *           T2,'ITER',
     *           T10, 'GRAVITY ANOMALY', T26,'GEOID HEIGHT',
     *           T40, 'GRAVITY ANOMALY', T56,'GEOID HEIGHT',
     *           T70, 'GRAVITY ANOMALY', T86,'GEOID HEIGHT',
     *           T100,'GRAVITY ANOMALY',T116,'GEOID HEIGHT'/
     *           T10, '     (MGAL)    ', T26,'   (METER)  ',
     *           T40, '     (MGAL)    ', T56,'   (METER)  ',
     *           T70, '     (MGAL)    ', T86,'   (METER)  ',
     *           T100,'     (MGAL)    ',T116,'   (METER)  '/
     *           T2,'----',
     *           T10,'---------------', T26,'------------',
     *           T40,'---------------', T56,'------------',
     *           T70,'---------------', T86,'------------',
     *          T100,'---------------',T116,'------------')
  621 FORMAT((1H0,/,T2,I3,T12,F9.4,T27,F9.4,T42,F9.4,T57,F9.4,
     *            T72,F9.4,T87,F9.4,T102,F9.4,T117,F9.4))
  622 FORMAT(1H1,T5,'THE DIFFERENCE BETWEEN THE PREDICTED AND ',
     *           ' THE MODEL GRAVITY ANOMALIES:',/,1H0)
  631 FORMAT(1H ,T5,'VALUES OF THE COVARIANCE TABLE USED',
     *           ' IN ITERATION: ',I3)
  632 FORMAT(1H ,T5,'VALUES OF THE CORRESPONDING POWER SPECTRA USED',
     *           ' IN ITERATION: ',I3)
  633 FORMAT(1X,I10,3D20.13)
  634 FORMAT(1H1,T5,'VALUES OF THE CORRESPONDING POWER SPECTRA USED',
```

```
     *                ' IN ITERATION: ',I3,/,1H0)
  635 FORMAT((1H ,5X,I10,3(5X,D20.13)))
  999 FORMAT(1H1,//,T10,'**** WRONG MODEL: ',I5,' USED. ****',//,
     *           T20,'APPROPRIATE MODELS ARE 36, 180 AND 300 !')
C
C
      STOP
      END
      SUBROUTINE POWSPC ( A, CV, X, P, LCT, LCTF, NFFT, IWK, WK )
C
C.....THIS SUBROUTINE DETERMINE THE POWER SPECTRUM OF A COVARIANCE
C       FUNCTION USING IMSL FFT SUBROUTINE
C
      REAL*8        A(LCTF), CV(LCT), P(LCTF), WK(NFFT)
      REAL*8        DI, DR, PMAX
      COMPLEX*16    X(LCTF)
      INTEGER       IWK(NFFT)
C
      PMAX = -1.0D0
      ND2 = LCTF/2 + 1
C
C.....FIRST SYMMETRIZE THE COVARIANCE FUNCTION
C
      DO 100  I = 1, LCT-1
        A(I) = CV(LCT+1-I)
        A(LCT-1+I) = CV(I)
  100 CONTINUE
C
C.....THEN SET THE END POINTS TO HAVE ZERO SLOPE
C
      A(1) = A(2)
      A(LCTF) = A(2)
C
      CALL FFTRC ( A, LCTF, X, IWK, WK )
C
C.....DETERMINE THE REMAINING COEFFICIENTS
C
      DO 110  I = 2, ND2
        X(LCTF+2-I) = DCONJG(X(I))
  110 CONTINUE
C
C.....DETERMINE THE POWER SPECTRUM
C
      DO 120  I = 1, LCTF
        DR = DREAL(X(I))
        DI = DIMAG(X(I))
        P(I) = DSQRT ( DR*DR + DI*DI )
  120 CONTINUE
C
C.....NORMALIZE THE COVARIANCE FUNCTION BEFORE THE FOURIER TRANSFORM
C
      DO 130  I = 1, LCTF
        PMAX = DMAX1 ( PMAX, P(I) )
  130 CONTINUE
C
      DO 140  I = 1, LCTF
        P(I) = P(I)/PMAX
  140 CONTINUE
C
      RETURN
      END
      SUBROUTINE EMPCVF ( D,CTV, M,N,DXY,MN, G,DG, H,DH, CG,CH,
     *                    CHHNS,CHHEW, CHGNS,CHGEW, CGGNS,CGGEW )
C
      REAL*4      CG(M,N), CH(M,N)
      REAL*4      G(M,N), DG(M,N), H(M,N), DH(M,N)
      REAL*4      CHHNS(0:MN-1), CHHEW(0:MN-1)
      REAL*4      CHGNS(0:MN-1), CHGEW(0:MN-1)
      REAL*4      CGGNS(0:MN-1), CGGEW(0:MN-1)
      REAL*8      D(MN), CTV(MN,3)
```

```
C
C.....REMOVE THE MODEL VALUE FROM THE DATA
C
      DO 100  I = 1, M
        DO 101  J = 1, N
          IF ( DH(I,J) .NE. -1000.0 .AND. H(I,J) .NE. -1000.0 )  THEN
            CH(I,J) = DH(I,J) - H(I,J)
          END IF
          IF ( DG(I,J) .NE. -1000.0 .AND. G(I,J) .NE. -1000.0 )  THEN
            CG(I,J) = DG(I,J) - G(I,J)
          END IF
  101   CONTINUE
  100 CONTINUE
C
C.....PERFORM THE CONVOLUTION
C
      DO 200  K = 0, MN-1
C
        SHH = 0.
        SHG = 0.
        SGG = 0.
        CHHNS(K) = 0.
        CHGNS(K) = 0.
        CGGNS(K) = 0.
        DO 210  J = 1, N
          DO 211  I = K+1, M
            IF ( CH(I-K,J).NE.-1000.0 .AND. CH(I,J).NE.-1000.0 )  THEN
              SHH = SHH + CH(I-K,J)*CH(I,J)
            END IF
            IF ( CH(I-K,J).NE.-1000.0 .AND. CG(I,J).NE.-1000.0 )  THEN
              SHG = SHG + CH(I-K,J)*CG(I,J)
            END IF
            IF ( CG(I-K,J).NE.-1000.0 .AND. CG(I,J).NE.-1000.0 )  THEN
              SGG = SGG + CG(I-K,J)*CG(I,J)
            END IF
  211     CONTINUE
  210   CONTINUE
        C1 = ((M-K)*N)
        CHHNS(K) = SHH / C1
        CHGNS(K) = SHG / C1
        CGGNS(K) = SGG / C1
C
        SHH = 0.
        SHG = 0.
        SGG = 0.
        CHHEW(K) = 0.
        CHGEW(K) = 0.
        CGGEW(K) = 0.
        DO 220  I = 1, M
          DO 221  J = K+1, N
            IF ( CH(I,J-K).NE.-1000.0 .AND. CH(I,J).NE.-1000.0 )  THEN
              SHH = SHH + CH(I,J-K)*CH(I,J)
            END IF
            IF ( CH(I,J-K).NE.-1000.0 .AND. CG(I,J).NE.-1000.0 )  THEN
              SHG = SHG + CH(I,J-K)*CG(I,J)
            END IF
            IF ( CG(I,J-K).NE.-1000.0 .AND. CG(I,J).NE.-1000.0 )  THEN
              SGG = SGG + CG(I,J-K)*CG(I,J)
            END IF
  221     CONTINUE
  220   CONTINUE
        C2 = (M*(N-K))
        CHHEW(K) = SHH / C2
        CHGEW(K) = SHG / C2
        CGGEW(K) = SGG / C2
C
        D(K+1) = DBLE ( K * DXY )
        CTV(K+1,1) = DBLE (( C1*CHHNS(K) + C2*CHHEW(K) ) / ( C1 + C2 ))
        CTV(K+1,2) = DBLE (( C1*CHGNS(K) + C2*CHGEW(K) ) / ( C1 + C2 ))
        CTV(K+1,3) = DBLE (( C1*CGGNS(K) + C2*CGGEW(K) ) / ( C1 + C2 ))
```

```
C
  200 CONTINUE
C
      RETURN
      END
      SUBROUTINE PREDIC ( IT, PVAL,VAR, EX,GY,GH, SGY,CGY, IREC,
     *                    SE, X1,Y1,SY1,CY1,
     *                    CM,CMS,CMI,JREC, IDIAG,CP,SRC,AP,
     *                    SD,CTV,CVUU,SCUU,CVUG,SCUG,CVGG,SCGG,
     *                    LCT, DTR, IER, NIER )
C
      REAL*8      SE(IREC)
      REAL*8      EX(IREC), GY(IREC), GH(IREC), SGY(IREC), CGY(IREC)
      REAL*8      CP(IREC), SRC(IREC), AP(IREC)
      REAL*8      CM(JREC), CMI(JREC), CMS(JREC)
      REAL*8      SD(LCT), CTV(LCT,3)
      REAL*8      CVUU(LCT), SCUU(LCT-1,3)
      REAL*8      CVUG(LCT), SCUG(LCT-1,3)
      REAL*8      CVGG(LCT), SCGG(LCT-1,3)
      REAL*8      D1, D2, CSD, DSD, VDSD, DTR
      REAL*8      X1, Y1, SY1, CY1, PVAL, VAR
      REAL*8      R1, R2
      INTEGER     IDIAG(IREC)
C
C.....RESET THE ERROR COUNTER
C
      IER = 0
C
      IF ( IT. EQ. 1 )  THEN
        R1 = CTV(1,1)
        R2 = CTV(1,3)
      ELSE IF ( IT. EQ. 2 )  THEN
        R1 = CTV(1,3)
        R2 = CTV(1,1)
      END IF
C
C.....CLEAR THE COVARIANCE MATRIX ARRAY
C
      CALL DCLR( CM, JREC )
      CALL DCLR( CMI, JREC )
      CALL DCLR( CMS, JREC )
C
      DO 100  I = 1, IREC
        CM(IDIAG(I)) = R1 + SE(I)*SE(I)
  100 CONTINUE
C
C.....EVALUATE ALL THE SINE AND COSINE TERMS
C
      DO 200  I = 1, IREC
        SGY(I) = DSIN(GY(I))
        CGY(I) = DCOS(GY(I))
  200 CONTINUE
C
C.....CONSTRUCT THE OFF-DIAGONAL TERMS OF THE SIGNAL COVARIANCE MATRIX
C
      DO 300  J = 2, IREC
        DO 301  I = 1, J-1
C
          CSD = SGY(I)*SGY(J) + CGY(I)*CGY(J)*DCOS(EX(I)-EX(J))
          DSD = DACOS(CSD)/DTR
          IF ( IT. EQ. 1 )  THEN
            CALL ICSEVU ( SD,CVUU,LCT, SCUU,LCT-1, DSD, VDSD, 1, IER )
          ELSE IF ( IT. EQ. 2 )  THEN
            CALL ICSEVU ( SD,CVGG,LCT, SCGG,LCT-1, DSD, VDSD, 1, IER )
          END IF
          K = (J*(J-1))/2 + I
          CM(K) = VDSD
C
  301   CONTINUE
  300 CONTINUE
```

```
C
C.....DUPLICATE THE COVARIANCE MATRIX FOR STORAGE
C
      DO 310  I = 1, JREC
        CMS(I) = CM(I)
  310 CONTINUE
C
C.....INVERT THE COVARIANCE MATRIX
C
      CALL LINV1P ( CMS,IREC,CMI, IDGT, D1,D2, IER )
C     CALL SOLVE ( CMS, SRC, IDIAG, IREC, .TRUE., .FALSE. )
C     CALL SYMINV ( CMS, SRC, IDIAG, IREC, CMI )
C
      IF ( IER. EQ. 0 )  THEN
C
C.......CONSTRUCT THE COVARIANCE VECTOR
C
        DO 330  I = 1, IREC
C
            CSD = SY1*SGY(I) + CY1*CGY(I)*DCOS(X1-EX(I))
            DSD = DACOS(CSD)/DTR
            CALL ICSEVU ( SD,CVUG,LCT, SCUG,LCT-1, DSD, VDSD, 1, IER )
            CP(I) = VDSD
C
  330   CONTINUE
C
C.......DETERMINE THE PREDICTED VALUE AT A GRID POINT
C
        CALL VMULSF ( CMI,IREC, GH,1,IREC, SRC,IREC )
C       CALL DCLR ( SRC, IREC )
C       CALL VSMXVT ( CMI, GH, SRC, IDIAG, IREC )
        PVAL = DOT ( CP, SRC, IREC )
C
C.......DETERMINE THE VARIANCE OF THE PREDICTED VALUE
C
        CALL VMULSF ( CMI,IREC, CP,1,IREC, AP,IREC )
        CALL VMULSF ( CM,IREC, AP,1,IREC, SRC,IREC )
C       CALL DCLR ( AP, IREC )
C       CALL DCLR ( SRC, IREC )
C       CALL VSMXVT ( CMI, CP, AP, IDIAG, IREC )
C       CALL VSMXVT ( CM, AP, SRC, IDIAG, IREC )
        VAR = R2 - 2.0D0*DOT(AP,CP,IREC) + DOT(AP,SRC,IREC)
C
        IF ( VAR. GT. 0.0 )  THEN
          VAR = DSQRT(VAR)
        ELSE
          IER = 1
          NIER = NIER + 1
        END IF
C
      END IF
C
      RETURN
      END
      FUNCTION DOT (A,B,N)
C
C  Compute the dot product of the two N-vectors A and B.
C
      INTEGER  N, I
      REAL*8   DOT
      REAL*8   A(1),B(1)
C
      DOT=0.0
C
      DO 100  I=1,N
  100 DOT=DOT + A(I)*B(I)
C
      RETURN
      END
      SUBROUTINE SYMINV ( A, B, IDIAG, NEQ, AINV )
```

```
C
      IMPLICIT REAL*8  (A-H,O-Z)
      DIMENSION    A(1), B(1), AINV(1), IDIAG(1)
C
      K = 0
      DO 100  I = 1, NEQ
        DO 101  L = 1, NEQ
          B(L) = 0.0D0
  101   CONTINUE
        B(I) = 1.0D0
        CALL SOLVE ( A, B, IDIAG, NEQ, .FALSE., .TRUE. )
        DO 110  J = 1, I
          K = K + 1
          AINV(K) = B(J)
  110   CONTINUE
  100 CONTINUE
C
      RETURN
      END
      SUBROUTINE SOLVE (A,B, IDIAG,NEQ, FACT,BACK)
C
C  Compute the U**T * D * U  factorization of the symmetric matrix
C  stored in A, if FACT = TRUE;  and solve A * X = B if BACK = TRUE.
C
C
C  A          Contains the compacted-column form of the upper triangular
C             part of the coefficient matrix.  After factorization, it
C             contains D and U.
C  B          Right-hand-side vector.  After backsubstitution, it
C             contains the solution.
C
C  IDIAG      Addresses of the diagonal terms in A.
C  NEQ        Number of equations
C
C  FACT       If FACT = TRUE, then factor A;  otherwise do not factor A.
C  BACK       If BACK = TRUE, reduce B and backsubstitute;  otherwise
C             do not solve the equations.
C
C
C
      IMPLICIT REAL*8  (A-H,O-Z)
      LOGICAL FACT,BACK
      DIMENSION A(1),B(1),IDIAG(1)
C
C
C  Factor A, reduce B
      JR = 0
      DO 400  J = 1, NEQ
        JD = IDIAG(J)
        JH = JD - JR
        IS = J - JH + 2
C
        IF (JH .LT. 2)  GOTO 390
C
        IF (FACT) THEN
C
          IF (JH .GT. 2) THEN
C..       Reduce column J rows IS to J-1: do not divide by row diagonal
            K = JR + 2
            ID = IDIAG(IS - 1)
C
            DO 100 I = IS, J-1
              IR = ID
              ID = IDIAG(I)
              IH = MIN (ID-IR-1,I-IS+1)
              IF (IH.GT.0)  A(K) = A(K) - DOT(A(K-IH),A(ID-IH),IH)
              K = K + 1
  100       CONTINUE
          ENDIF
C
```

```
C..        Divide by row diagonal, and reduce diagonal term in column J
C
           IR = JR + 1
           K = J - JD
           DO 200  I = IR, JD-1
             ID = IDIAG(K+I)
             IF (A(ID).EQ.0.0) GOTO 200
             D = -A(I)
             A(I) = A(I)/A(ID)
             A(JD) = A(JD) + D*A(I)
  200      CONTINUE
C
         ENDIF
C
C..      Reduce RHS
         IF (BACK)  B(J) = B(J) - DOT(A(JR+1),B(IS-1),JH-1)
C
  390    JR = JD
  400  CONTINUE
C
C
       IF (.NOT.BACK)  RETURN
C
C
C  Divide by diagonal pivots
C
       DO 700 I = 1,NEQ
         ID = IDIAG(I)
         IF (A(ID).NE.0.0)    B(I) = B(I)/A(ID)
  700  CONTINUE
C
C  Backsubstitute
C
       J = NEQ
       JD = IDIAG(J)
C
  801  D = -B(J)
       J = J - 1
       IF (J.LE.0)  RETURN
C
       JR = IDIAG(J)
       IF (JD-JR.GT.1) THEN
         IS = J - JD + JR + 2
         K = JR - IS + 1
         DO 810  I = IS, J
  810    B(I) = B(I) + A(I+K)*D
       ENDIF
C
       JD = JR
       GOTO 801
C
C
       END
       SUBROUTINE VSMXVT ( A, B, C, JDIAG, NEQ )
C
C.....THIS SUBROUTINE FORMS C = C + A*B WHERE A IS A SYMMETRIC MATRIX
C       STORED IN PROFILE FORM, B, C ARE VECTORS, AND JDIAG LOCATES THE
C       DIAGONALS IN A.
C
       IMPLICIT REAL*8  (A-H,O-Z)
       DIMENSION   A(1), B(1), C(1), JDIAG(1)
C
       JS = 1
       DO 200  J = 1, NEQ
         JD = JDIAG(J)
         IF ( JS. LE. JD )  THEN
           BJ = B(J)
           AB = A(JD)*BJ
           IF ( JS. NE. JD )  THEN
             JB = J - JD
```

```
            JE = JD - 1
            DO 100  JJ = JS, JE
              AB = AB + A(JJ)*B(JJ+JB)
              C(JJ+JB) = C(JJ+JB) + A(JJ)*BJ
  100       CONTINUE
          END IF
          C(J) = C(J) + AB
        END IF
        JS = JD + 1
  200 CONTINUE
C
      RETURN
      END
      SUBROUTINE DCLR (A,NA)
C
      INTEGER  NA, I
      REAL*8   A(1)
C
      DO 100  I = 1, NA
        A(I) = 0.0D0
  100 CONTINUE
C
      RETURN
      END
//*
// EXEC LINKGOV,REGION=5000K
//*
//SYSLIB  DD DSN=SYS2.IMSLD,DISP=SHR
//*       DD DSN=SYS2.IMSLS,DISP=SHR
//*
//* **** **** **** **** **** **** **** **** **** **** **** *
//*                                                        *
//*    BLACK SEA GRIDDED COLLOCATION GEOIDAL HEIGHT DATA   *
//*                                                        *
//* **** **** **** **** **** **** **** **** **** **** **** *
//*
//GO.FT05F001 DD DSN=ZMAYA.ALTIM.DATA(TABLE1),DISP=SHR
//GO.FT06F001 DD SYSOUT=*
//GO.FT08F001 DD DSN=ZMAYA.BLACKC4H.GRID,DISP=SHR
//GO.FT09F001 DD DSN=ZMAYA.BLACKC4H.COVM,DISP=SHR
//GO.FT10F001 DD DSN=ZMAYA.ALTIM.DATA(BRH180),DISP=SHR
//GO.FT11F001 DD DSN=ZMAYA.ALTIM.DATA(BRG180),DISP=SHR
//GO.FT12F001 DD DSN=ZMAYA.ALTIM.DATA(BRH36),DISP=SHR
//GO.FT13F001 DD DSN=ZMAYA.ALTIM.DATA(BRG36),DISP=SHR
//GO.FT14F001 DD DSN=ZMAYA.ALTIM.DATA(BRH300),DISP=SHR
//GO.FT15F001 DD DSN=ZMAYA.ALTIM.DATA(BRG300),DISP=SHR
//GO.FT16F001 DD DSN=ZMAYA.BLACK.MASK,DISP=SHR
//GO.FT17F001 DD DSN=ZMAYA.ALTIM.DATA(BSGAB0R),DISP=SHR
//GO.FT18F001 DD DSN=ZMAYA.ALTIM.COVM(BSGAB0R),DISP=SHR
//GO.FT19F001 DD DSN=ZMAYA.ALTIM.DATA(BSGUB0R),DISP=SHR
//GO.FT20F001 DD DSN=ZMAYA.ALTIM.COVM(BSGUB0R),DISP=SHR
//GO.FT21F001 DD DSN=ZMAYA.ALTIM.DIFF(BSGUB0R),DISP=SHR
//GO.FT22F001 DD DSN=ZMAYA.ALTIM.CVTB(B0R),DISP=SHR
//*
// EXEC NOTIFYTS
```

A listing of PROGRAM JORDAN

```
//ZMAYABJC JOB (G0109,360,2),AYAU,TIME=(00,30),CLASS=O,MSGCLASS=X
// EXEC FORTVC
//SYSIN DD *
C
      PARAMETER ( SN =  1.27 )
      PARAMETER ( SG = 11.20 )
      PARAMETER ( D =  0.38 )
      PARAMETER ( DR = 0.25 )
      PARAMETER ( RMIN = 0.0 )
      PARAMETER ( RMAX = 10.0 )
      PARAMETER ( SN2 = SN*SN )
      PARAMETER ( SG2 = SG*SG )
      PARAMETER ( L = RMAX/DR + 1 )
C
      REAL*8      R(L), CNN(L), CNG(L), CGG(L)
      REAL*8      RD, RDS, RD2, EXPRD, BI0, BI1, BK0, BK1
      REAL*8      DSN2, DSG2, DD, DDR, CNST
      REAL*8      MMBSI0, MMBSI1, MMBSK0, MMBSK1
      INTEGER     OUP, OUC
C
      DATA        OUP / 6 /, OUC / 8 /
C
      DSN2 = DBLE(SN2)
      DSG2 = DBLE(SG2)
      DDR = DBLE(DR)
      DD = DBLE(D)
      CNST = DBLE(2.*SN*SG)/DSQRT(6.0D0)
C
      R(1) = DBLE(RMIN)
      DO 100  I = 2, L
        R(I) = R(I-1) + DDR
  100 CONTINUE
C
      CNN(1) = DSN2
      CGG(1) = DSG2
      CNG(1) = CNST
      DO 200  I = 2, L
C
      RD = R(I)/DD
      RDS = RD*RD
      RD2 = RD/2.0D0
      EXPRD = DEXP(-RD)
      BI0 = MMBSI0(1,RD2,IER)
      BI1 = MMBSI1(1,RD2,IER)
      BK0 = MMBSK0(1,RD2,IER)
      BK1 = MMBSK1(1,RD2,IER)
C
      CNN(I) = DSN2 * ( 1.0D0 + RD + RDS/3.0D0 ) * EXPRD
C
      CGG(I) = DSG2 * ( 1.0D0 + RD - RDS/2.0D0 ) * EXPRD
C
      CNG(I) = CNST * ( RD2*(1.0D0-(RDS/2.0D0))*(BI0*BK1-BI1*BK0)
     *                + (RDS/4.0D0)*(BI0*BK0+BI1*BK1) )
C
  200 CONTINUE
C
      WRITE ( OUP, 600 )
      WRITE ( OUP, 601 )  (R(I),CNN(I),CNG(I),CGG(I),I=1,L)
      WRITE ( OUC, 801 )  (R(I),CNN(I),CNG(I),CGG(I),I=1,L)
C
  600 FORMAT(1H1,//,T5,'THE COVARIANCE FUNCTION C(NN), C(NG) & C(GG):',
     *           //)
  601 FORMAT((1H ,5X,D10.4,3(5X,D20.13)))
  801 FORMAT((1X,D10.4,3D20.13))
C
      STOP
      END
```

```
//*
// EXEC LINKGOV,REGION=5000K
//*
//SYSLIB   DD DSN=SYS2.IMSLD,DISP=SHR
//*        DD DSN=SYS2.IMSLS,DISP=SHR
//*
//GO.FT06F001 DD SYSOUT=*
//GO.FT08F001 DD DSN=ZMAYA.ALTIM.DATA(BJ180),DISP=SHR
//*
// EXEC NOTIFYTS
```

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-100730		

4. Title and Subtitle

User's Guide
Programs for Processing Altimeter Data Over Inland Seas

5. Report Date

April 1989

6. Performing Organization Code

621

7. Author(s)

A.Y. Au, R.D. Brown and J.E. Welker

8. Performing Organization Report No.

89B00106

10. Work Unit No.

RTOP #676

9. Performing Organization Name and Address

ST Systems Corporation (STX)
4400 Forbes Boulevard
Lanham, Maryland 20706

11. Contract or Grant No.

13. Type of Report and Period Covered

Technical Memorandum

12. Sponsoring Agency Name and Address

Goddard Space Flight Center
National Aeronautics and Space Administration
Washington, D.C. 20546-0001

14. Sponsoring Agency Code

15. Supplementary Notes (continued from block 7)

A.Y. Au and R.D. Brown: ST Systems Corporation (STX), Lanham, Maryland.
J.E. Welker: Goddard Space Flight Center, Greenbelt, Maryland.

16. Abstract

Satellite-based altimetric data taken by GEOS-3 and SEASAT over the Black Sea and Caspian Sea have been analyzed and a least-squares collocation technique used to predict the geoid undulations on a .25-degree by .25-degree grid and to transform these geoid undulations to free air gravity anomalies. The programs used to process and analyze these altimeter data are described and documented herein. A brief user's guide is presented for each program that summarizes the purpose for, input to, and output from the code. This is followed by a complete listing of well commented code in the FORTRAN-77 language. These programs read, organize, and plot profiles and ground tracks of the altimeter data. They also identify crossovers between altimeter passes, determine and remove biases from the data, sort, and grid the corrected altimetric sea surface heights. Other programs generate contour plots of the gridded data, determine the gravity and geoid auto- and cross-covariance functions, and use these functions in a collocation estimate of gravity anomalies.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Black Sea, Caspian Sea, gravity anomalies, satellite altimetry	Unclassified - Unlimited Subject Category 46

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified		

PREPARATION OF THE REPORT DOCUMENTATION PAGE

The last page of a report facing the third cover is the Report Documentation Page, RDP. Information presented on this page is used in announcing and cataloging reports as well as preparing the cover and title page. Thus it is important that the information be correct. Instructions for filling in each block of the form are as follows:

Block 1. Report No. NASA report series number, if preassigned.

Block 2. Government Accession No. Leave blank.

Block 3. Recipient's Catalog No. Reserved for use by each report recipient.

Block 4. Title and Subtitle. Typed in caps and lower case with dash or period separating subtitle from title.

Block 5. Report Date. Approximate month and year the report will be published.

Block 6. Performing Organization Code. Leave blank.

Block 7. Author(s). Provide full names exactly as they are to appear on the title page. If applicable, the word editor should follow a name.

Block 8. Performing Organization Report No. NASA installation report control number and, if desired, the non-NASA performing organization report control number.

Block 9. Performing Organization Name and Address. Provide affiliation (NASA program office, NASA installation, or contractor name) of authors.

Block 10. Work Unit No. Provide Research and Technology Objectives and Plans (RTOP) number.

Block 11. Contract or Grant No. Provide when applicable.

Block 12. Sponsoring Agency Name and Address. National Aeronautics and Space Administration, Washington, D.C. 20546-0001. If contractor report, add NASA installation or HQ program office.

Block 13. Type of Report and Period Covered. NASA formal report series; for Contractor Report also list type (interim, final) and period covered when applicable.

Block 14. Sponsoring Agency Code. Leave blank.

Block 15. Supplementary Notes. Information not included elsewhere: affiliation of authors if additional space is required for block 9, notice of work sponsored by another agency, monitor of contract, information about supplements (film, data tapes, etc.), meeting site and date for presented papers, journal to which an article has been submitted, note of a report made from a thesis, appendix by author other than shown in block 7.

Block 16. Abstract. The abstract should be informative rather than descriptive and should state the objectives of the investigation, the methods employed (e.g., simulation, experiment, or remote sensing), the results obtained, and the conclusions reached.

Block 17. Key Words. Identifying words or phrases to be used in cataloging the report.

Block 18. Distribution Statement. Indicate whether report is available to public or not. If not to be controlled, use "Unclassified-Unlimited." If controlled availability is required, list the category approved on the Document Availability Authorization Form (see NHB 2200.2, Form FF427). Also specify subject category (see "Table of Contents" in a current issue of STAR), in which report is to be distributed.

Block 19. Security Classification (of this report). Self-explanatory.

Block 20. Security Classification (of this page). Self-explanatory.

Block 21. No. of Pages. Count front matter pages beginning with iii, text pages including internal blank pages, and the RDP, but not the title page or the back of the title page.

Block 22. Price Code. If block 18 shows "Unclassified-Unlimited," provide the NTIS price code (see "NTIS Price Schedules" in a current issue of STAR) and at the bottom of the form add either "For sale by the National Technical Information Service, Springfield, VA 22161-2171" or "For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402-0001," whichever is appropriate.