

TR 76-3

# Storm Surge Simulation in Transformed Coordinates

## VOLUME II Program Documentation

by

John J. Wanstrath

TECHNICAL REPORT NO. 76-3  
NOVEMBER 1976



Approved for public release;  
distribution unlimited.

Prepared for  
U.S. ARMY, CORPS OF ENGINEERS  
**COASTAL ENGINEERING  
RESEARCH CENTER**

Kingman Building  
Fort Belvoir, Va. 22060

Reprint or republication of any of this material shall give appropriate credit to the U.S. Army Coastal Engineering Research Center.

Limited free distribution within the United States of single copies of this publication has been made by this Center. Additional copies are available from:

*National Technical Information Service  
ATTN: Operations Division  
5285 Port Royal Road  
Springfield, Virginia 22151*

Contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

## UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR 76-3	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) STORM SURGE SIMULATION IN TRANSFORMED COORDINATES VOLUME I. THEORY AND APPLICATION VOLUME II. PROGRAM DOCUMENTATION		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John J. Wanstrath, Robert E. Whitaker, Robert O. Reid, and Andrew C. Vastano		8. CONTRACT OR GRANT NUMBER(s) DACPW72-73-C-0014
9. PERFORMING ORGANIZATION NAME AND ADDRESS Texas A&M Research Foundation F.E. Box H College Station, Texas 77843		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS A31231
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Army Coastal Engineering Research Center Kingman Building, Fort Belvoir, Virginia 22060		12. REPORT DATE November 1976
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES Vol. I. 166 Vol. II. 176
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Computer program Hurricane Gracie Coordinate transformation Numerical modeling Hurricane Camille Orthogonal curvilinear coordinates Hurricane Carla Storm surge		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A two-dimensional time-dependent numerical storm surge model using orthogonal curvilinear coordinates is presented. The curvilinear coordinate system is based on a conformal mapping of the interior region bounded by the actual coast, the seaward boundary (taken as the 180-meter depth contour) and two parallel lateral boundaries into a rectangle in the image plane. Three regions of the Continental Shelf of the Gulf of Mexico and two regions of the eastern seaboard of the United States are mapped.		

(Continued)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Since the transformation is conformal, the associated modifications of the vertically integrated equations of motion and mass continuity are minimized. The coast, seaward boundary, and the lateral boundaries of the computing grid are straight lines in the image plane thus facilitating the application of the boundary conditions. The final coordinates allow for the greatest resolution near the coast in a central area of principal storm surge development and modification.

The model is employed in the simulation of the storm surge induced by Hurricanes Carla (1961) and Camille (1969) which crossed the gulf coast of the United States and Hurricane Gracie (1959) which crossed the east coast. Analytical interpretations of the wind and atmospheric pressure-forcing functions are used in the computations.

## PREFACE

This report is published to provide coastal engineers with the results of a study to develop an operational program for numerical simulation of storm surges on a given segment of the Continental Shelf, using a curvilinear coordinate system. The report consists of two volumes. Volume I discusses the theory and application of the transformation procedure for generating the curvilinear shelf coordinate system for particular regions, and the theory, numerical algorithm, and application of the storm surge program for simulation of Hurricanes Carla (1961), Camille (1969), and Gracie (1959). Volume II presents the program documentation and the coded programs for carrying out the coordinate transformation (CONFORM), for establishing the spatial lattice (GRID), and for carrying out the storm surge calculations on the shelf (SSURGE). The work was carried out under the wave mechanics program of the U.S. Army Coastal Engineering Research Center (CERC).

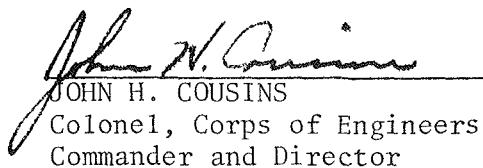
This volume was prepared by John J. Wanstrath; Volume I was prepared by John J. Wanstrath, Robert E. Whitaker, Robert O. Reid, and Andrew C. Vastano, Department of Oceanography, Texas A&M University, College Station, Texas, under CERC Contract No. DACW72-73-C-0014. Most of the computational work in the development and application was carried out at the National Center for Atmospheric Research which is supported by the National Science Foundation.

The authors express their appreciation to Thomas J. Reid for assistance in program coding, and to Dr. D. Lee Harris, CERC, for very constructive comments on the draft of this report.

Dr. D. Lee Harris, Chief, Oceanography Branch, was the CERC technical monitor of the report, under the general supervision of Mr. R.P. Savage, Chief, Research Division.

Comments on this publication are invited.

Approved for publication in accordance with Public Law 166, 79th Congress, approved 31 July 1945, as supplemented by Public Law 172, 88th Congress, approved 7 November 1963.



JOHN H. COUSINS  
Colonel, Corps of Engineers  
Commander and Director

## CONTENTS

		Page
	CONVERSION FACTORS, U.S. CUSTOMARY TO METRIC (SI) . . . . .	6
I	INTRODUCTION. . . . .	7
II	COMPUTER PROGRAM DOCUMENTATION FOR PROGRAM-CONFORM. . . . .	8
	1. Program Purpose . . . . .	8
	2. Program Description . . . . .	8
	3. Type of Computer. . . . .	9
	4. Input Data. . . . .	9
	5. Mathematical Procedures and Program Limitations . . . . .	10
	6. Flow Chart. . . . .	11
	7. Glossary and Description of Terms . . . . .	15
	8. Input and Output. . . . .	17
III	COMPUTER PROGRAM DOCUMENTATION FOR PROGRAM-GRID . . . . .	20
	1. Program Purpose . . . . .	20
	2. Program Description . . . . .	20
	3. Type of Computer. . . . .	21
	4. Input Data. . . . .	22
	5. Mathematical Procedures and Program Limitations . . . . .	22
	6. Flow Chart. . . . .	25
	7. Description of Terms. . . . .	31
	8. Input and Output. . . . .	33
IV	COMPUTER PROGRAM DOCUMENTATION FOR PROGRAM-SSURGE . . . . .	35
	1. Program Purpose . . . . .	35
	2. Program Description . . . . .	35
	3. Type of Computer. . . . .	36
	4. Input Data. . . . .	36
	5. Mathematical Procedures and Program Limitations . . . . .	38
	6. Flow Chart. . . . .	39
	7. Glossary and Description of Terms . . . . .	41
	8. Input and Output. . . . .	44
APPENDIX		
A	FORTRAN LISTING OF PROGRAM CONFORM. . . . .	68
B	FORTRAN LISTING OF PROGRAM GRID . . . . .	113
C	FORTRAN LISTING OF PROGRAM SSURGE . . . . .	150

## CONVERSION FACTORS, U. S. CUSTOMARY TO METRIC (SI) UNITS OF MEASUREMENT

U.S. customary units of measurement used in this report can be converted to metric (SI) units as follows:

Multiply	by	To obtain
inches	25.4	millimeters
	2.54	centimeters
square inches	6.452	square centimeters
cubic inches	16.39	cubic centimeters
feet	30.48	centimeters
	0.3048	meters
square feet	0.0929	square meters
cubic feet	0.0283	cubic meters
yards	0.9144	meters
square yards	0.836	square meters
cubic yards	0.7646	cubic meters
miles	1.6093	kilometers
square miles	259.0	hectares
acres	0.4047	hectares
foot-pounds	1.3558	newton meters
ounces	28.35	grams
pounds	453.6	grams
	0.4536	kilograms
ton, long	1.0160	metric tons
ton, short	0.9072	metric tons
degrees (angle)	0.1745	radians
Fahrenheit degrees	5/9	Celsius degrees or Kelvins <sup>1</sup>

<sup>1</sup> To obtain Celsius (C) temperature readings from Fahrenheit (F) readings, use formula:  $C = (5/9)(F - 32)$ .  
To obtain Kelvin (K) readings, use formula:  $K = (5/9)(F - 32) + 273.15$ .

## STORM SURGE SIMULATION IN TRANSFORMED COORDINATES

### Volume II. Program Documentation

by  
*John J. Wanstrath*

#### I. INTRODUCTION

Presented in this volume are the program documentation and listings of the coded programs for carrying out a simulation of a hurricane-induced storm surge on the Continental Shelf in curvilinear coordinates for a given reach of shelf. There are three separate programs detailed here for this purpose.

The first program, CONFORM, is employed for evaluation of the transformation coefficients which map the given reach of the Continental Shelf into a rectangle in the image plane, the shelf region being delineated by a smoothed version of the coastline, the shelf break (taken as the 180-meter depth contour in the examples), and bounded laterally by two parallel straight lines. The theory and several example applications of the transformation procedure are given in Section II of Volume I. The example input for CONFORM contained in the listings here are for the mapping of the region from a section across Laguna Madre about 90 kilometers south of Brownsville to Marsh Island. Particular care must be taken to follow the procedure exactly for the example if CONFORM is to be verified by obtaining the given transformation coefficients. This procedure is given explicitly in the CONFORM documentation.

The second program, GRID, develops the detailed computing grid information, based on the mapping coefficients evaluated by CONFORM plus coordinate stretching information supplied by the user (see Section III of Volume I). Part of the output of GRID is a listing of the grid positions which are required by the user in order to read from appropriate charts the detailed bathymetry field, which is necessary input for the final program SSURGE. The example data supplied here for GRID are for the Laguna Madre to Marsh Island region.

Program SSURGE (Shelf Surge) carries out the numerical integration of the storm surge equations in the transformed coordinate system supplied by CONFORM and GRID, using a parametric representation of a hurricane wind field and pressure field. The theory is given in Section III of Volume I. The particular example data given here are for Hurricane Carla and the Laguna Madre-Marsh Island grid system.

The appendixes to this volume contain detailed FORTRAN listings of the three programs in this application. The data to be supplied by the user for other applications are discussed in the documentation of each of these programs.

## II. COMPUTER PROGRAM DOCUMENTATION FOR PROGRAM-CONFORM

### 1. Program Purpose.

The purpose of the program is to determine the transformation coefficients which will conformally map the interior region bounded by the actual coastline, a seaward boundary curve, and two parallel lateral boundaries into a rectangle in the image plane.

### 2. Program Description.

The program is written in FORTRAN IV language. This program and the program GRID provide all the necessary computing grid data for input to program SSURGE. The program GRID takes, as input, the transformation coefficients and determines the computing grid information (such as, scale factors, grid point locations, and, at each grid point, the orientation of the  $\xi$ -axis to the x-axis).

The program CONFORM is composed of:

MAIN	Defines constants. Reads and writes the coordinates delineating the given coastline and seaward boundary curve. Calls Subroutine COEFFS.
SUBROUTINE COEFFS	Determines the transformation coefficients. At the completion of each iteration, the coefficients, the variance between the transform-generated curves and that specified, and other pertinent information are written. COEFFS interfaces all other program subroutines and functions.
FUNCTION XTRAN	Is the transformation function $x(\xi, \eta)$ .
FUNCTION YTRAN	Is the transformation function $y(\xi, \eta)$ .
SUBROUTINE SLFAC	Determines the scale factor and derivatives, $\partial x / \partial \xi$ , $\partial y / \partial \xi$ , for a given value of $\xi$ and $\eta$ .
SUBROUTINE CUR1YB	Determines the necessary parameters to fit a spline under tension through the given <u>coastline</u> coordinates. The spline is fitted with $Y_2$ as a function of $X_2$ .
FUNCTION CURVYB	Interpolates the given <u>coastline</u> , returning a value of $y$ at a specified value for $x$ .

FUNCTION CURDYB	Differentiates the given <u>coastline</u> , returning a value of $dy/dx$ at a specified value for $x$ .
SUBROUTINE CUR2YB	Determines the necessary parameters to fit a spline under tension with $X2$ as a function of <u>coastline arclength</u> .
FUNCTION CUR4YB	Interpolates the given <u>coastline</u> returning a value of $x$ at a specified value for arclength.
SUBROUTINE CUR3YB	Determines the necessary parameters to fit a spline under tension with $Y2$ as a function of <u>coastline arclength</u> .
FUNCTION CUR5YB	Interpolates the given <u>coastline</u> returning a value of $y$ at a specified value for arclength.

There are identical subroutines and functions as delineated above for the seaward boundary curve specified by coordinates  $X2P$  and  $Y2P$ . These subroutines and functions are recognized by the same names as their counterparts with a terminal letter A. For example, SUBROUTINE CUR1YA determines the necessary parameters to fit a spline under tension through the given seaward boundary curve.

### 3. Type of Computer.

The program CONFORM can be run on any computer with minimum core requirements of approximately 24K (based on the present sample program). However, significantly more computer memory would be required if one desires a large number of coefficients and/or numerous integration points. The program has been executed successfully on IBM 360, CDC/6600 and 7600, and GE/635. The present sample program requires no auxiliary storage devices, peripheral devices, or magnetic tape input or output. No site-orientated computer plot routines are involved in the program. Approximately 20 minutes of machine time on a CDD/7600 is required for the sample program (total number of coefficients,  $2 \times NMAX = 220$ ; number of integration points,  $0 < \xi < \lambda$ ,  $\lambda = 110$ ; and number of iterations = 80).

### 4. Input Data.

Input data, other than constants defined in MAIN, are read in MAIN on IBM cards prepared according to the following list:

#### (1) Card 1

IWANT, MQ, MOP, NMAX, Continuation flag, number of shore-  
JMAS1, IL, VARWT line and seaward boundary coordinates

number of mapping coefficients, maximum number of iterations, number of integration points, and the convergence criterion in format 6I5, F5.3.

(2) Card Group 2

X2,Y2      The  $x,y$  coordinates (units in  $x,y$  space) of the given coastline in the region  $0 \leq x \leq \lambda$  are read with one pair per card in format 3X, F7.2, 3X, F7.2 (limit 150).

(3) Card Group 3

X2P,Y2P      The  $x,y$  coordinates (units in  $x,y$  space) of the given seaward boundary curve in the region  $0 \leq x \leq \lambda$  are read with one pair per card in format F7.2, 3X, F7.2 (limit 150).

Optional      Card 4, Card Group 3, and Card 6

If IWANT = 1, indicating the program is being restarted, the following cards must be supplied:

(4) Card 4

B, BZRO      The values of  $\beta$  and  $B_0$  in units of length of  $x,y$  space from the last iteration of the previous run in format 2E14.7.

(5) Card Group 5

COB,COC      The NMAX cards containing the dimensionless Fourier-type transformation coefficients from the last iteration of the previous run in sequential order with one pair per card (format 2E14.7). If more coefficients are desired in the present run than the previous one, blank cards should be supplied for the difference.

(6) Card 6

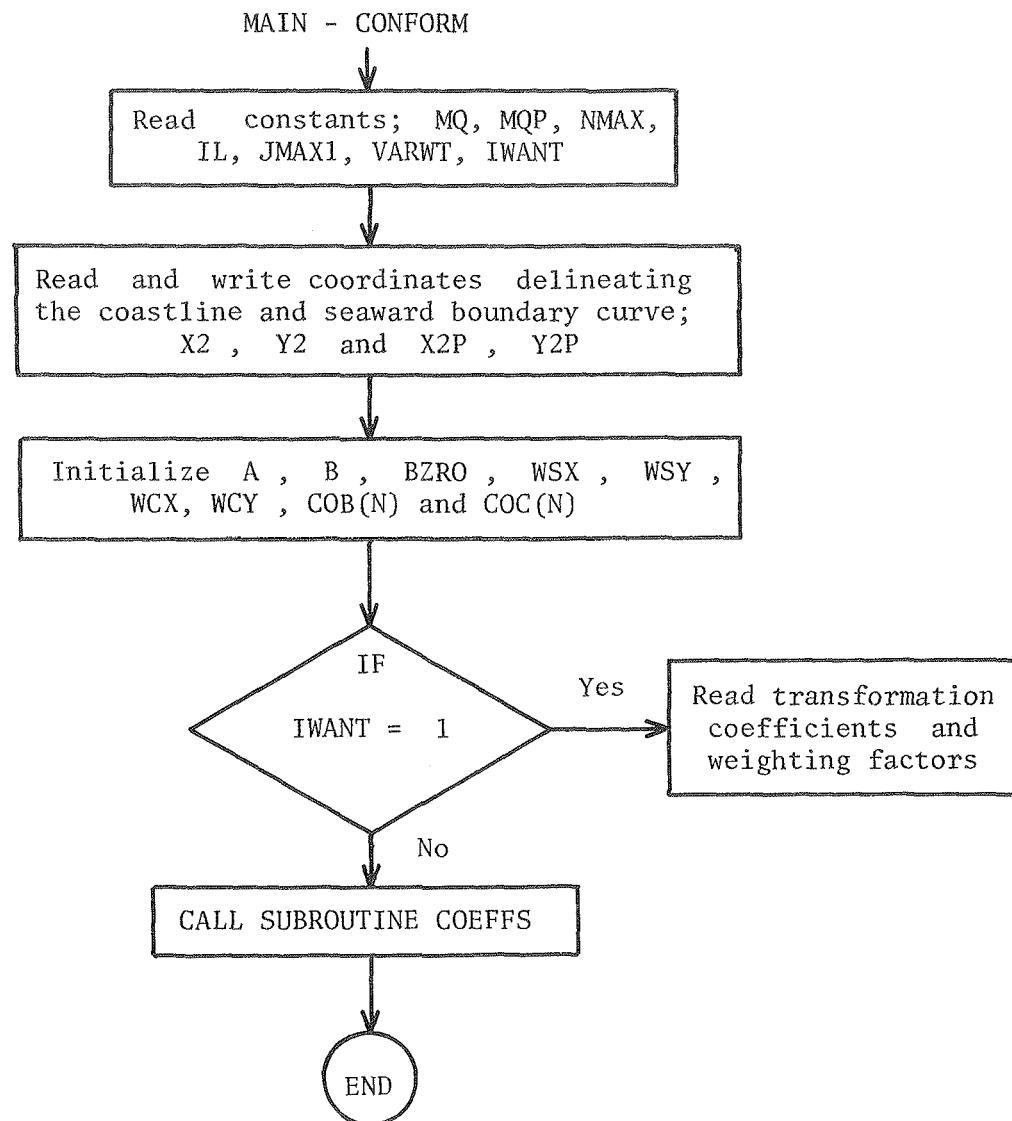
WSX,WSY      The value of the weighting factors for the seaward boundary curve ( $x$  and  $y$  component) and coastline ( $x$  and  $y$  component) from the last iteration of the previous run (4E14.7).

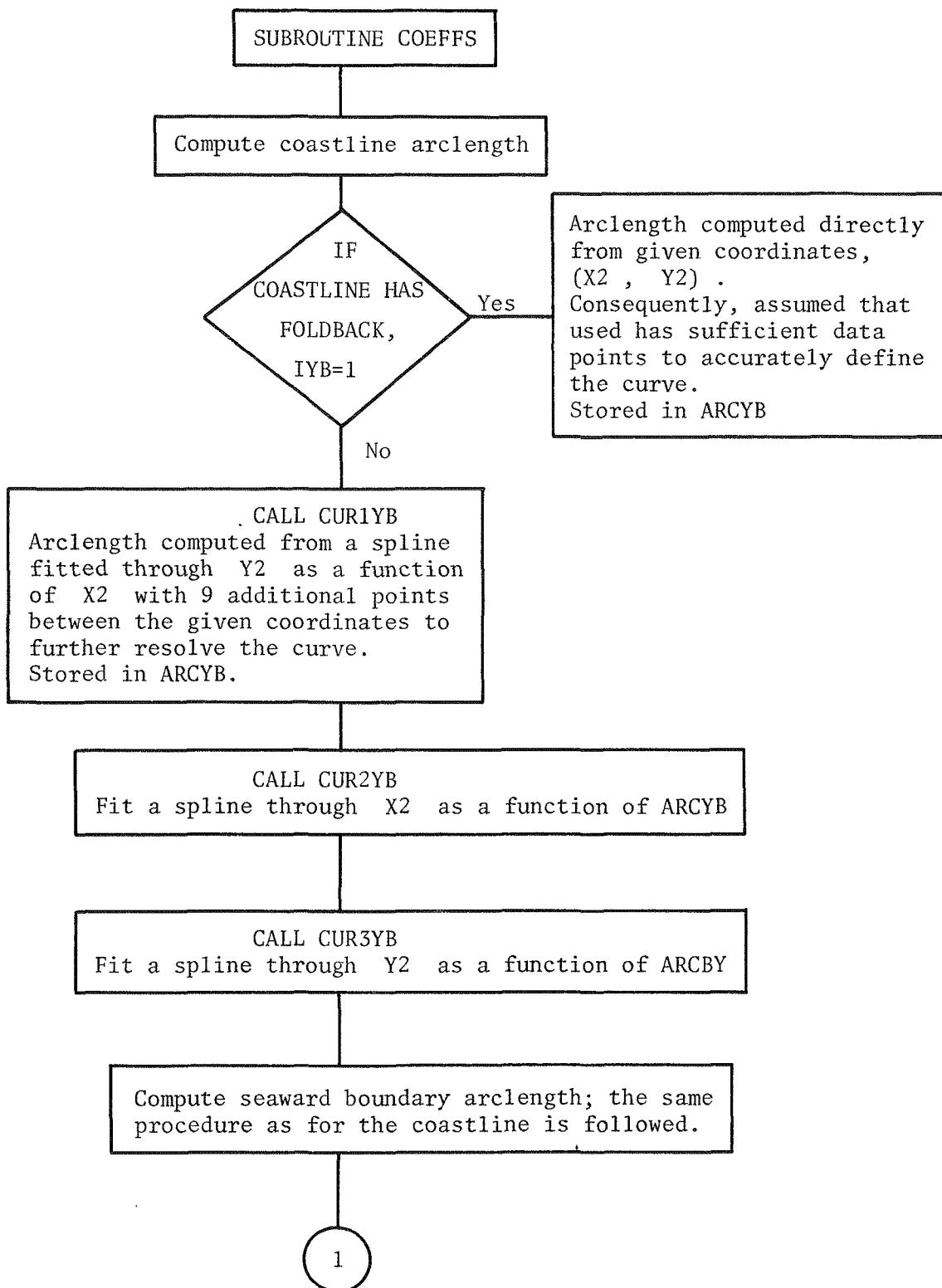
5. Mathematical Procedures and Program Limitations.

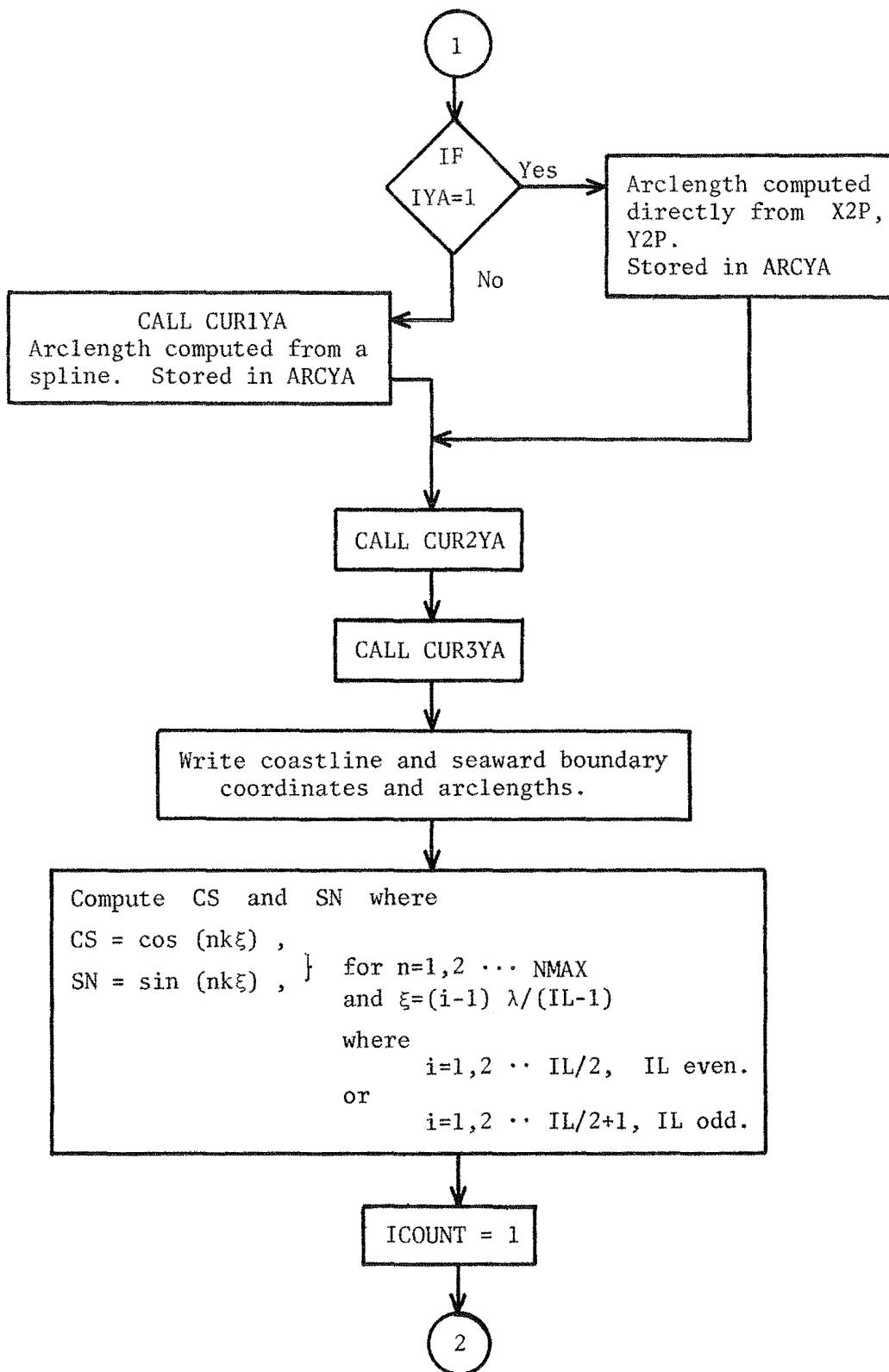
The conformal mapping relations, solutions for the transformation coefficients, and the iterative procedure for determining the coefficients are presented in Volume I of this report. The mapping equations are sufficiently general to treat the situation where either or

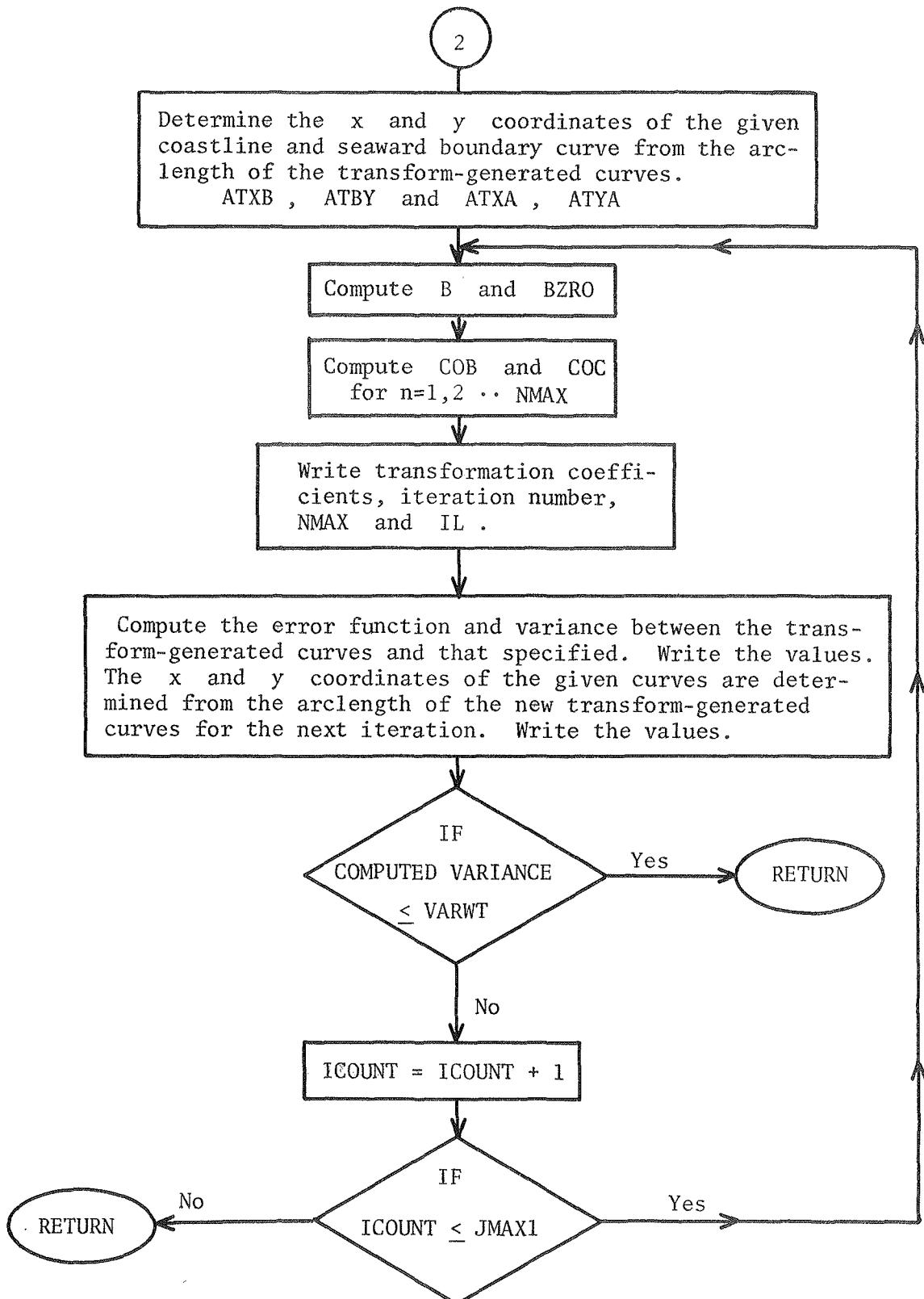
both given curves are multivalued in  $y$  for a specified  $x$  value. The only program limitation relates to the computer memory capacity. No program error messages or consistency checks are incorporated in this routine.

6. Flow Chart.









## 7. Glossary and Description of Terms.

### Arrays

X2	Dimensioned MQ. Contains values of the x coordinate in units of x,y space of the given coastline (max. 150).
Y2	Dimensioned MQ. Contains values of the y coordinate in units of x,y space of the given coastline (max. 150).
X2P	Dimensioned MQP. Contains values of the x coordinate in units of x,y space of the given seaward boundary curve (max. 150).
Y2P	Dimensioned MQP. Contains values of the y coordinate in units of x,y space of the given seaward boundary curve (max. 150).
COB	Dimensioned NMAX. Contains values of the dimensionless transformation coefficient $B_n$ (max. 200).
COC	Dimensioned NMAX. Contains values of the dimensionless transformation coefficient $C_n$ (max. 200).
DUMB	Dimensioned NMAX. Contains the iterative value for $B_n$ (max. 200).
DUMC	Dimensioned NMAX. Contains the iterative value for $C_n$ (max. 200).
ARCYB	Dimensioned MQ. Contains values of the arclength for the given coastline coordinates X2 , Y2 (max. 150).
ARCYA	Dimensioned MQP. Contains values of the arclength for the given seaward boundary coordinates X2P , Y2P (max. 150).
SN	Dimensioned NMAX $\times$ IL1 where IL1 is IL/2 for even IL or IL1 is IL/2 + 1 for odd IL. Contains values of $\sin(nk\xi)$ .
CS	Dimensioned NMAX $\times$ IL1. Contains values of $\cos(nk\xi)$ .
ATXB	Dimensioned IL. Contains values of the x coordinate of the given coastline as determined from the arclength of the transform-generated curve (max. 400).
ATYB	Dimensioned IL. Contains values of the y coordinate of the given coastline as determined from the arclength of the transform-generated curve (max. 400).

ATXA Dimensioned IL. Contains values of the x coordinate of the given seaward boundary curve as determined from the arclength of the transform-generated curve (max.400).  
 ATYA Dimensioned IL. Contains values of the y coordinate of the given seaward boundary curve as determined from the arclength of the transform-generated curve (max.400).  
 Z Dimensioned IL. Contains values of  $\xi$  (max.400).  
 R,S,T,U,  
 V,W,X,Y Dimensioned IL. Temporary storage.

#### Constants

MQ Number of coordinates delineating the given coastline (max.150).  
 MQP Number of coordinates delineating the given seaward boundary curve (max.150).  
 NMAX Number of transformation coefficients,  $B_n$  or  $C_n$  (max.200).  
 IL Number of equally spaced integration points for  $0 \leq \xi \leq \lambda$  with  $IL \geq NMAX$  (max.400).  
 IWANT If IWANT = 1, the program is to be re-started requiring input from the previous run. If IWANT  $\neq 1$ , it is the initial mapping of the region.  
 VARWT Desired variance (in units of x,y space squared) between the transform-generated curves and that specified.  
 XLAMDA  $\lambda = X2(MQ) = X2P(MQP)$   
 XK  $k = \pi/\lambda$   
 A -B  
 B  $\beta$   
 BZRO  $B_0$   
 WSX,WSY The x and y component of the weighting factor for the seaward boundary curve.  
 WCX,WCY The x and y component of the weighting factor for the coastline.

## 8. Input and Output.

The input data required by CONFORM to determine the transformation coefficients which conformally map the Laguna Madre, Mexico, to Marsh Island, Louisiana, region into a rectangle are presented here as an example. The first card image gives the continuation code, number of coastline and seaward boundary coordinates, number of coefficients desired, maximum number of iterations, number of integration points, and the convergence criterion. The next 47 paired numbers are the coastline coordinates and the last 40 card images give the seaward boundary coordinates.

Program CONFORM provides detailed and voluminous output concerning primarily the rate of convergence. These output statements are not necessary for program completion and can be easily deleted with little alteration to the sequence of instructions. Optional output statements are indicated in the program listing (App. A) by an arrow ( $\leftarrow$ ) on the right-hand side of the page.

The results from CONFORM required by Program GRID are  $\beta$ ,  $B_n^0$ , and the coefficients,  $B_n^0$  and  $C_n^0$ . These are given in the next section as input to program GRID.

Reference to Table 4 in Volume I of this report shows the values for the number of coefficients, maximum number of iterations, and number of integration points are only indicative of the final steps of this particular application. Explicitly, in order to obtain the given coefficients to conformally map the Laguna Madre to Marsh Island region into a rectangle in the image plane, the following steps must be followed:

- (1) Set NMAX to 40 and IL to 80 for the first 20 iterations.
- (2) For the next 10 iterations, NMAX is 60 and IL is 120.
- (3) Set NMAX to 80 and IL to 160 for iterations 31 through 50.
- (4) Take NMAX as 90 and IL as 180 for the next 10 iterations.
- (5) For iterations 61 through 70, NMAX is 100 and IL is 200.
- (6) Over the next 10 iterations, NMAX is 110 and IL is 220.
- (7) Starting with the 81st iteration, WCX and WSX are set to 0 and NMAX is 110 and IL is 220 through the 100th iteration.

- (8) From iteration 101 through 110, NMAX is 130 and IL is 260.
- (9) Over the last 10 iterations, NMAX is 150 and IL is 300.

Note that steps (7) through (9) utilize the alternate solution to the mapping equations (9) and (10) in Section II of Volume I of this report. The instructions which must be altered or removed are indicated in the program listing by parenthesis with the proper instruction enclosed.

The following input is required by CONFORM to conformally map the Laguna Madre, Mexico to Marsh Island, Louisiana region into a rectangle in the image plane. Note the fourth and fifth parameters on the first card image are only indicative of the final results. See Input and Output text for explanation of procedure used to obtain the mapping coefficients given as input to Program GRID.

```

C
C      I N P U T      D A T A
C
C      1    47    40   150   1500.001           360.00     110.00
C
C      COASTLINE COORDINATES          C      SEAWARD BOUNDARY COORDINATES
C
C
000.00    035.00           000.00    002.00
005.60    040.00           007.50    005.00
010.50    044.80           016.00    010.00
018.00    051.00           021.50    014.50
028.00    059.00           028.00    019.00
040.00    066.80           036.00    023.00
050.50    073.00           045.00    028.00
056.50    076.50           054.00    032.00
063.00    082.00           060.00    036.00
070.00    093.00           066.00    040.00
074.00    102.00           077.00    046.00
076.50    108.50           085.00    051.00
079.00    116.00           089.00    055.00
082.50    127.00           095.00    064.00
088.00    137.80           104.00    076.00
095.00    146.00           110.00    084.00
106.00    154.00           112.00    089.00
121.00    162.00           113.50    095.00
136.00    167.00           114.50    101.00
153.00    168.30           117.50    108.00
158.00    168.00           121.00    112.00
161.50    167.30           127.00    116.00
166.00    168.80           139.00    121.00
171.00    169.00           153.00    123.50
183.00    168.00           165.00    123.00
197.00    166.00           179.00    119.00
214.00    165.00           193.00    115.00
230.00    166.00           213.00    104.00
242.00    167.00           229.00    092.00
250.00    166.70           244.00    080.00
253.40    167.20           257.00    071.00
257.00    169.50           267.00    066.00
265.00    169.00           278.00    062.00
280.00    166.00           289.00    056.00
291.00    163.00           298.00    049.00
296.00    159.30           307.00    042.00
298.50    159.90           318.00    035.00
303.00    159.00           333.00    027.00
314.00    153.00           347.00    020.00
323.00    148.00           360.00    014.00
328.00    142.00
331.00    137.20
335.50    129.20
342.50    121.00
349.50    115.00
352.50    113.00

```

### III. COMPUTER PROGRAM DOCUMENTATION FOR PROGRAM-GRID

#### 1. Program Purpose.

The purpose of this program is to determine the grid point array in the stretched curvilinear shelf coordinate system and appropriate scale factors needed for program SSURGE. The detailed grid is needed in order for the user to read off depths from an appropriate bathymetric chart of the shelf region at grid locations.

#### 2. Program Description.

The program is written in FORTRAN IV language. This program interfaces between Programs CONFORM and SSURGE. It is assumed that the conformal mapping of the storm surge region has been completed to the user's satisfaction. The program GRID takes, in part, as input, the transformation coefficients and determines computing grid information of scale factors, grid point locations, and, at each grid point, the orientation of the  $\xi$ -axis to the x-axis .

The program GRID is composed of

MAIN	Defines constants. Reads transformation coefficients outputed from CONFORM. Reads water depths along a line near center of grid from the seaward boundary to the coast. Computes grid point locations, scale factors ( $u$ , $v$ , and $F$ ), and, at each grid point, $\cos \theta$ and $\sin \theta$ . Writes computing grid information.
SUBROUTINE XUT	Writes information transferred into XUT.
SUBROUTINE SHCOR	Determines and writes grid point coordinates in x,y space and the distance in nautical miles between points.
SUBROUTINE TRAN	Computes the x and y coordinates of the transform-generated coastline and seaward boundary curve.
SUBROUTINE TRAN1	Computes $x(\xi, \eta)$ and $y(\xi, \eta)$ .
SUBROUTINE TRAN2	Computes $\partial x / \partial \xi$ , $\partial y / \partial \xi$ and $\theta = \tan^{-1} (\frac{\partial y}{\partial x} / \frac{\partial \xi}{\partial x})$ .
SUBROUTINE CURV9	Contains the expansion curve $Y = Z + B(X^C)$ where A , B , and C are constants. The term Y is either Sp (units, nautical miles) or T (units, minutes). The term X is either S* (units, nautical miles) or T* (units, minutes). This subroutine computes Y and $dY/dX$ given the coefficients and X .

- SUBROUTINE CURV1 - Determines the necessary parameters to compute an interpolatory spline under tension through a sequence of functional values contained in arrays X2 and Y2 .
- FUNCTION CURV2 - Interpolates the given curve, Y2 as a function of X2 , returning a value for y given x .
- SUBROUTINE CURV3 - Determines the necessary parameters to compute an interpolatory spline under tension through a sequence of functional values contained in arrays X2P and Y2P.
- FUNCTION CURV4 - Interpolates the given curve, Y2P as a function of X2P , returning a value for y given x .

### 3. Type of Computer.

The program GRID may be run on any computer with minimum core requirements of approximately 26K words (based on the present sample program appropriate to the Hurricane Carla surge simulation grid). GRID requires no auxiliary storage devices, peripheral devices, or magnetic tape input or output. No site-oriented computer plot routines are involved in the program. Approximately 25 minutes of machine time on a GE/635 is required for the sample program to determine the computing grid information. This time is based on the following pertinent program parameters:

- a) 150 transformation coefficients,  $B_n$  or  $C_n$  ;
- b) 121 evenly spaced values of  $\xi$  for determining the transform-generated coastline arclength as a function of  $\xi$  ;
- c) 51 evenly spaced values of  $\eta$  for determining the arclength along a particular isoline of  $\xi$  as a function of  $\eta$  ;
- d) the computing grid of 47  $\xi$  (or  $S^*$ ) lines and 15  $\eta$  (or  $T^*$ ) lines;
- e) for determining the scale factor  $F$  , the area in  $x,y$  space of each quadrangle is approximated by using 4 evenly spaced increments between  $\xi$  isolines and 2 evenly spaced increments between  $\eta$  isolines.

For production runs, smaller sampling intervals might be required in b, c and especially, c.

#### 4. Input Data.

Input data, other than constants defined in MAIN, are read in MAIN and are on IBM cards prepared according to the following list:

##### (1) Card 1

NMAX, NUMXI, Number of mapping coefficients, number of  $\xi$   
NUMETA, DELSS, lines + 2, number of  $\eta$  lines,  $\Delta S^*$  in nautical  
SSTRT, DELTT, miles, first value of  $S^*$  in nautical miles,  
ND, NS  $\Delta T^*$  in minutes, number of depths, number of  
points in format 314, 3F5.1, 214.

##### (2) Card 2

BETA BZRO The value of  $\beta$  and  $B_0$  from the conformal  
mapping solution in format 2E14.7.

##### (3) Card Group 3

COB,COC The NMAX cards containing the Fourier-type  
transformation coefficients,  $B_n$  and  $C_n$ ,  
in format 2E14.7.

##### (4) Card Group 4

SY Temporary storage for the ND values of the water  
depth (fathoms) along a line from the seaward  
boundary to the coast. This information is  
needed to evaluate the travelttime coordinate T.

#### 5. Mathematical Procedures and Program Limitations.

Information concerning the expanding grid procedure and the  
relations transforming  $\xi, \eta$  to  $S^*, T^*$  space is presented in  
Volume I of this report. The user is required to know the coeffi-  
cients of the expansion function

$$S_p = A + B(S^*)^C$$

for each region of the curve where  $S_p$  is arclength (nautical miles)  
along the transform-generated coastline. For the sample program,  
there are five regions of the expansion curve. Selecting  $\Delta S^* =$   
6 nautical miles, the number of  $\Delta S^*$  intervals of each region and  
the value of  $\partial S_p / \partial S^*$  at the end points of each region, we can  
determine the coefficients of each region from three simultaneous  
equations derived from the constraints:

For region I,  $176 \text{ nautical miles} \leq S^* \leq 236 \text{ nautical miles}$   
(10 intervals of  $\Delta S^*$  )

$$\frac{\partial S_p}{\partial S^*} = 2.6225 \quad \text{at } S^* = 176 \text{ nautical miles}$$

$$\frac{\partial S_p}{\partial S^*} = 1.5 \quad \text{at } S^* = 236 \text{ nautical miles}$$

$$S_p = S_p \quad \text{at } S^* = 236 \text{ nautical miles}$$

For region II,  $236 \text{ nautical miles} \leq S^* \leq 260 \text{ nautical miles}$   
(4 intervals of  $\Delta S^*$  )

$$\frac{\partial S_p}{\partial S^*} = 1.5 \quad \text{at } S^* = 236 \text{ nautical miles}$$

$$\frac{\partial S_p}{\partial S^*} = 1.5 \quad \text{at } S^* = 260 \text{ nautical miles}$$

$$S_p = S_p \quad \text{at } S^* = 260 \text{ nautical miles}$$

For region III,  $260 \text{ nautical miles} \leq S^* \leq 302 \text{ nautical miles}$   
(7 intervals of  $\Delta S^*$  )

$$\frac{\partial S_p}{\partial S^*} = 1.5 \quad \text{at } S^* = 260 \text{ nautical miles}$$

$$\frac{\partial S_p}{\partial S^*} = 1.0 \quad \text{at } S^* = 302 \text{ nautical miles}$$

$$S_p = 302 \text{ nautical miles at } S^* = 302 \text{ nautical miles}$$

For region IV,  $302 \text{ nautical miles} \leq S^* \leq 356 \text{ nautical miles}$   
(9 intervals of  $\Delta S^*$  )

$$\frac{\partial S_p}{\partial S^*} = 1.0 \quad \text{at } S^* = 302 \text{ nautical miles}$$

$$\frac{\partial S_p}{\partial S^*} = 1.0 \quad \text{at } S^* = 356 \text{ nautical miles}$$

$$S_p = 302 \text{ nautical miles at } S^* = 302 \text{ nautical miles}$$

For region V,  $356$  nautical miles  $\leq S^* \leq 452$  nautical miles  
(16 intervals of  $\Delta S^*$ )

$$\frac{\partial S_p}{\partial S^*} = 1.0 \quad \text{at } S^* = 356 \text{ nautical miles}$$

$$\frac{\partial S_p}{\partial S^*} = 2.7 \quad \text{at } S^* = 452 \text{ nautical miles}$$

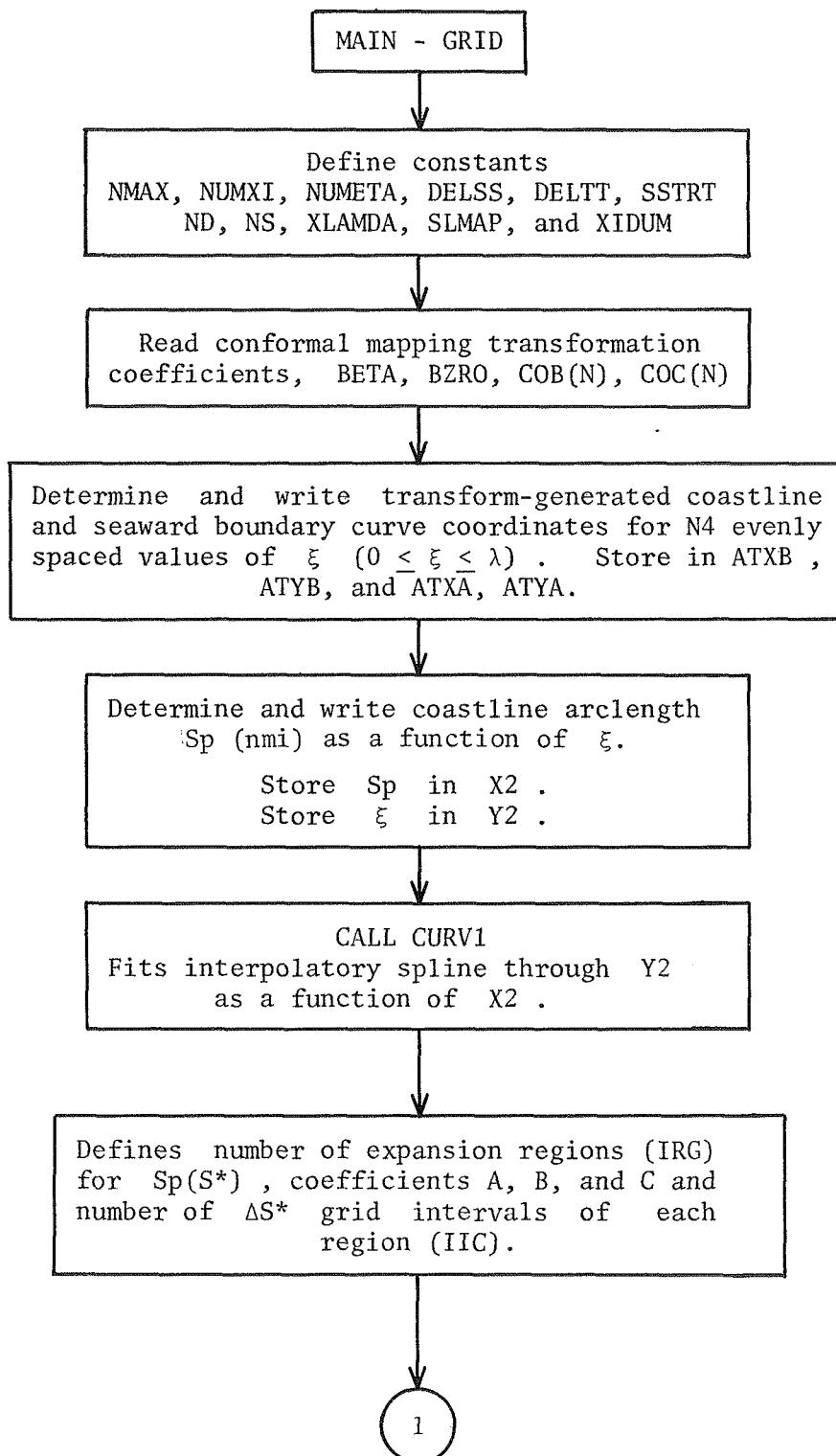
$$S_p = 356 \text{ nautical miles at } S^* = 356 \text{ nautical miles}$$

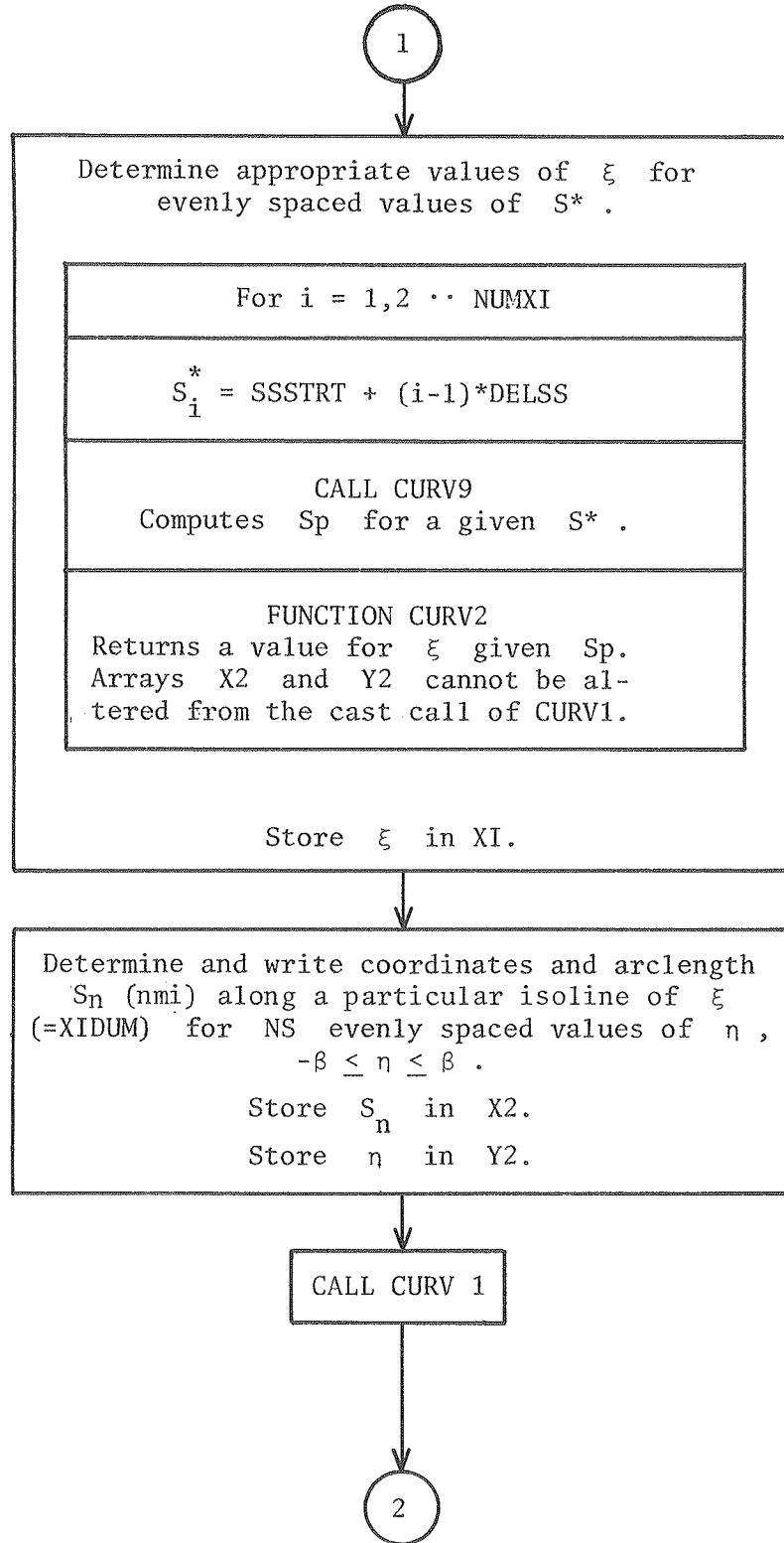
A similar procedure is followed for the expansion function

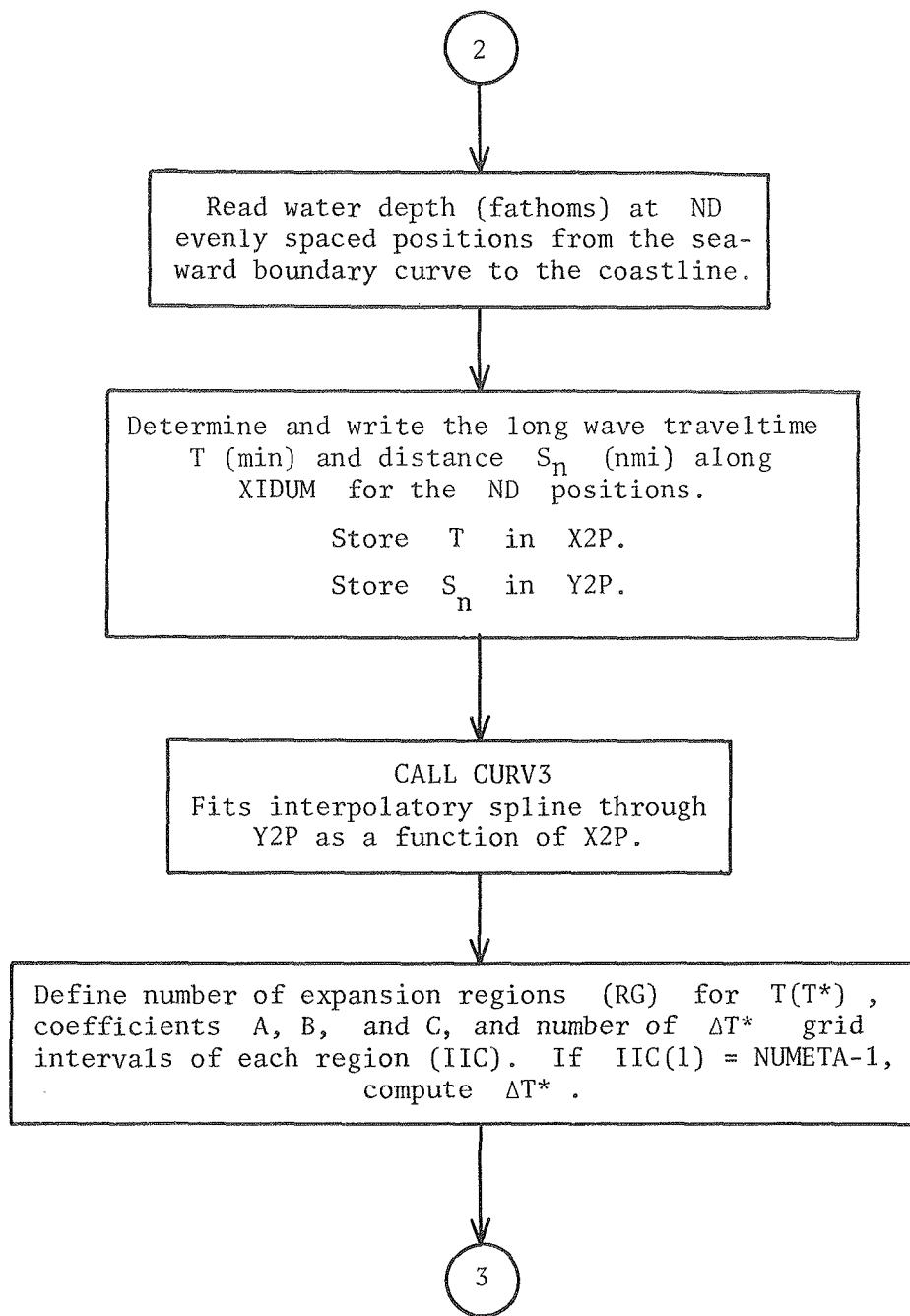
$$T = A + B(T^*)^C$$

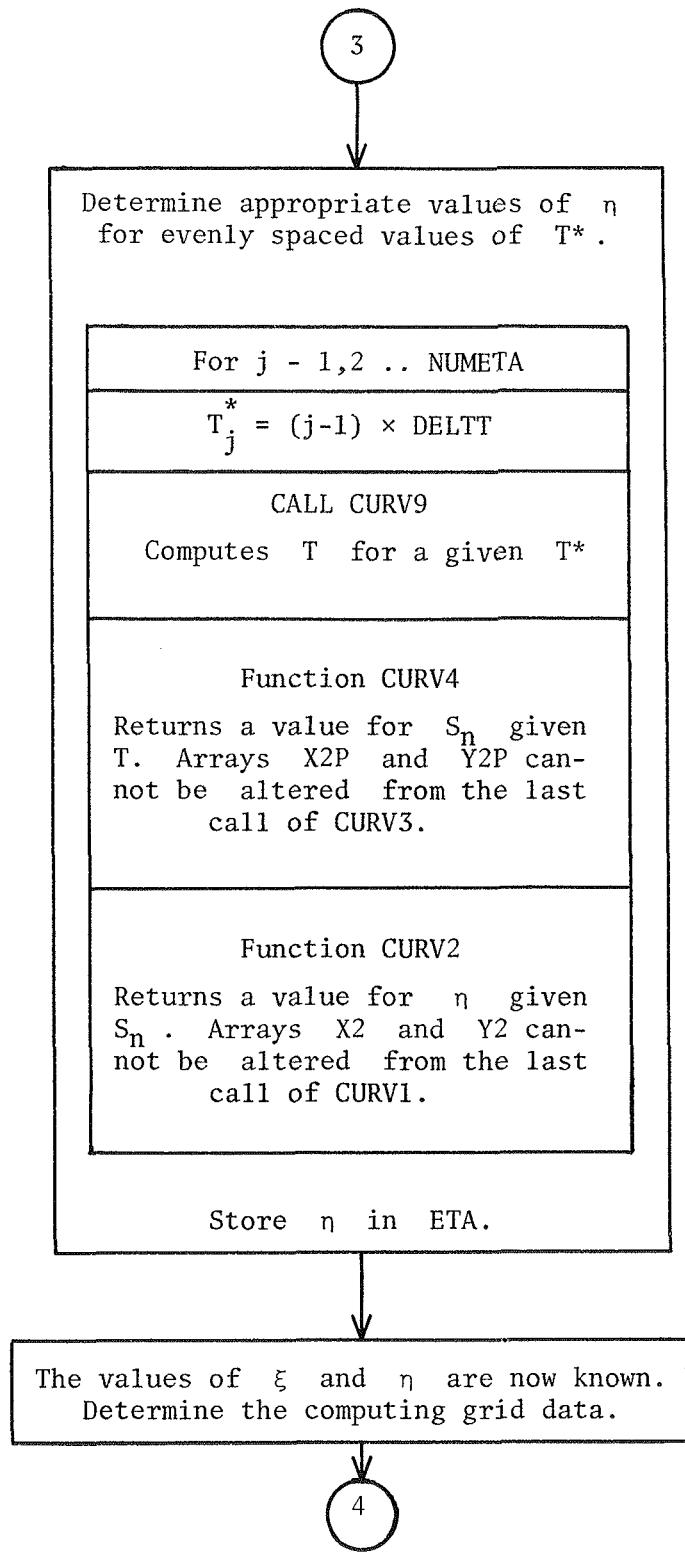
where  $T$  is the long wave traveltimes (minutes) along the particular isoline of  $\xi$ . The program assumes that there are, at most, two regions of the expansion curve with the second region being  $\Delta T = \Delta T^*$ . The expansion coefficients are determined by the program. If the user desires only one region (i.e.,  $T = T^*$  and  $\Delta T^* = \text{total long wave traveltimes-number of } n \text{ grid intervals}$ ), the program computes  $\Delta T^*$ .

6. Flow Chart.









4

CALL SHRCOR

Determines and writes the grid point coordinates  
in x,y space and the distance (nmi) between  
grid points.

Store x coordinate in X and  
Store y coordinate in Y for  
 $i = 2,3 \dots NUMXI-1$   
 $j = 1,2 \dots NUMETA.$

Determine and write scale factor  $\mu$  (units  
x,y space-nmi).

Store  $\mu$  in SX for  $i=1,2 \dots NUMXI-2$ .

Determine and write scale factor  $\nu$  (units of  
x,y space-time minutes).

Store  $\nu$  in SY for  $j=1,2 \dots NUMETA.$

Determine and write dimensionless scale factor F.  
Compute area in x,y plane of each quadrangle  
subdivided into IQUAD intervals between  $\xi$   
lines and JQUAD intervals between  $\eta$  lines.

Store area in X for  $i=1,2 \dots NUMXI-1$   
 $j=1,2 \dots NUMETA.$

Store F in Y for  $i=1,2 \dots NUMXI-2$   
 $j=1,2 \dots NUMETA.$

5

5

Determine and write  $\cos \theta$  and  $\sin \theta$  where  
 $\theta$  is the orientation of the  $\xi$  axis to  
the x axis.

Store  $\theta$  in Z for  $i=1,2 \dots \text{NUMXI}$   
 $j=1,2 \dots \text{NUMETA}.$

Store  $\cos \theta$  in X and  
 $\sin \theta$  in Y for  $i=1,2 \dots \text{NUMXI-2}$   
 $j=1,2 \dots \text{NUMETA}.$

END

## 7. Description of Terms.

### Arrays

All arrays except COB, COC, XI and ETA are reused throughout the program. The user is cautioned to consult each major program division for assessment of array contents.

X, Y, Z	Dimensioned NUMXI × NUMETA.
X2, Y2	
X2P, Y2P	
ATXA, ATYA	
ATXB, ATYB	Dimensioned the larger of N4, NS, ND, or IQUAD.
A, B, C	
IIC	Dimensioned IRG (the number of Sp(S*) expansion regions).
SX	Dimensioned NUMXI-2.
SY	Dimensioned NUMETA.
COB	Dimensioned NMAX. Contains the conformal mapping transformation coefficients, $B_n$ .
COC	Dimensioned NMAX. Contains the conformal mapping transformation coefficients, $C_n$ .
XI	Dimensioned NUMXI. Contains the values of $\xi$ for determining the computing grid data.
ETA	Dimensioned NUMETA. Contains the values of $\eta$ for determining the computing grid data.

### Constants

NMAX	Number of conformal mapping transformation coefficients, $B_n$ or $C_n$ .
NUMXI	Number of computing grid $\xi$ lines. With respect to the computing grid in Program SSURGE, there is an extra $\xi$ line at each lateral end. This requirement results from the averaging procedure used in determining the grid data.
NUMETA	Number of computing grid $\eta$ lines. This is the same number as in Program SSURGE.

DELSS	The value in nautical miles of $\Delta S^*$ . This corresponds to DXI in Program SSURGE.
	Since in SSURGE the product, $\mu \Delta S^*$ , is always computed in the surge equations, we determine DXI in meters such that $\mu$ values are dimensionless, i.e.,
	$DXI = 1852 \frac{m}{nmi} \times \Delta S^* \text{ nmi} \times \left[ SLMAP \frac{nmi}{x,y \text{ unit}} \times \mu \text{ unit} \right]$
DELT T	The value in minutes at $\Delta T^*$ . This corresponds to DETA in Program SSURGE.
	Since in SSURGE, the product, $v \Delta T^*$ , is always computed in the surge equations, we determine DETA in meters such that $v$ values are dimensionless, i.e.,
	$DETA = 1852 \frac{m}{nmi} \times \Delta T^* \text{ min} \times \left[ SLMAP \frac{nmi}{x,y \text{ unit}} \times v \text{ unit} \right]$
SSSTRT	The first value of $S^*$ in nautical miles.
ND	The number of water depths (fathoms) inputed from the seaward boundary to the coast for determining distance as a function of long wave travelttime.
NS	The number of points along XIDUM for determining $\eta$ as a function of arclength $S_n$ .
XLAMDA	Horizontal extent of the mapped region in units of x,y space.
XIDUM	The particular value of $\xi$ used in determining $\eta(S_n)$ .
SLMAP	The chart scale relating distance in nautical miles to distance in x,y units (i.e., nmi is equivalent to 51 units of length in x,y space).
G	Acceleration due to gravity (feet.s <sup>-2</sup> ).
N4	The number of points used in determining $\xi$ as a function of arclength $S_p$ .
IRG	The number of expansion regions of $SP(S^*)$ or $t(T^*)$ .

IQUAD      The area in  $x,y$  space of each grid quadrangle is subdivided into IQUAD intervals between  $\xi$  lines and  
and JQUAD intervals between  $\eta$  lines.

#### 8. Input and Output.

The first card input to Program GRID gives the number of conformal transformation coefficients NMAX , the number of  $\xi$  lines NUMXI, the number of  $\eta$  lines NUMETA,  $\Delta S^*$  in nautical miles DELSS, the first value of  $S^*$  in nautical miles SSSTRT,  $\Delta T^*$  in minutes DELTT, the number of water depths ND , and the number of points US used to establish  $\eta = \eta(S_n)$  .

The second card gives  $\beta$  and  $B_0$  and the next 150 cards give the mapping coefficients  $B_n$  and  $C_n$  . These 151 cards are the punched output from Program CONFORM.

The remaining 31 cards are the depths picked off a bathymetric chart of the northwestern Gulf of Mexico. These depths are on a constant  $\xi$ -line selected by the user.

Expansion coefficients, provided by the user, appear as statements within the program after format 135.

The reader must refer to Section III of Volume I of this report for an explanation of the parameters associated with the stretched shelf coordinate system for the Hurricane Carla surge simulation.

Output from GRID consists of the transform-generated coastline and seaward boundary coordinates, the transform-generated arclengths along the coastline and seaward boundary, and for each of the five sections,  $\partial S_p / \partial S^*$  ,  $S^*$  ,  $S_p$  , and  $\xi$  are listed. Additionally, the transform-generated arclengths at values of  $\eta$  for evenly spaced increments of  $T^*$  and at  $\eta$  values for constant increments of  $T$  along the chosen  $\xi$ -line, the  $\eta$  values along the specified  $\xi$ -line such that  $\Delta T^*$  is constant, and the traveltimes and depths along the constant  $\xi$ -line are printed.

The following output is required by Program SSURGE for simulating the Hurricane Carla surge. This includes the  $x,y$  coordinates of the grid intersections, the scale factors  $\mu$  and  $\nu$  related to the transformation of  $\xi$  to  $S^*$  and  $\eta$  to  $T^*$  , respectively, the scale factor  $F$  , and the sines and cosines of theta giving the orientation of the  $\xi$ -axis to the  $x$ -axis. The program listing indicates when these may be punched or written on tape or disk for convenient input to SSURGE.



## IV. COMPUTER PROGRAM DOCUMENTATION FOR PROGRAM-SSURGE

### 1. Program Purpose.

The purpose of the program is to numerically simulate the storm surge in orthogonal curvilinear coordinates with the vertically integrated form of the quasi-linear long-wave equations.

### 2. Program Description.

The program is written in FORTRAN IV language. It is assumed that the conformal mapping of the region under investigation has been completed. The transformation coefficients for three regions of the continental shelf of the Gulf of Mexico and two regions of the eastern seaboard are provided in Appendixes A and E in Volume I of this report.

An interfacing program is required which inputs the coefficients and generates a curvilinear computing grid to the user's satisfaction. The output from the program (and, in part, also the input to SSURGE) must be the scale factors, grid point locations, and, at each grid point, the orientation of the  $\xi$ -axis to the x-axis.

The Program SSURGE is composed of:

MAIN	Defines constants and interfaces the subroutines.
SUBROUTINE ZERO	Initializes all arrays to zero.
SUBROUTINE FIELD	Reads data and writes the water depth field relative to mean sea level, the wind field parameters and the storm positions.
SUBROUTINE WINDF	Calculates the wind and atmospheric pressure fields.
SUBROUTINE ELEV	Computes the water level anomaly, H .
SUBROUTINE FLUX	Computes transports, $Q_S^*$ and $Q_T^*$ .
SUBROUTINE DRAW1	Outputs H and vertically averaged water velocities, $Q_S^*/\bar{D}$ and $Q_T^*/\bar{D}$ , at hourly time intervals and saves the water level anomaly at prescribed grid locations for output at program completion.

SUBROUTINE METER	Calculates and saves the vertically averaged water velocities at prescribed grid locations for output at program completion.
SUBROUTINE HUV	Outputs the saved information of the simulated hydrographs, simulated current meters, and observed water levels at program completion.

### 3. Type of Computer.

The program SSURGE may be run on any computer with minimum core requirements of approximately 30K words of memory (based on the present sample program appropriate to the Hurricane Carla surge simulation). The program has been executed successfully on the IBM 360, CDC/6600 and 7600, and GE/635. The present sample program requires no auxiliary storage devices, peripheral devices or magnetic tape input or output. No site-dependent computer plot routines are involved in the program. Approximately 14.4 minutes of machine time on a GE/635 is required for the sample program to complete 66 hours of surge simulation.

### 4. Input Data.

Input data, other than constants defined in MAIN, are read in SUBROUTINE FIELD. These data are on cards prepared according to the following list.

#### (1) Card 1

NT1 - Number of cards (max. 50) containing on each TIM, ROT, RAD, VRMAX and PZRO (format I5).

#### (2) Card Group 2

NT1 cards with each card containing values of

- a) TIM      The time in hours at which the hurricane wind and atmospheric pressure parameters are recorded (format F10.1).
- b) ROT      The angle in degrees between the direction the storm is moving and the region of maximum winds (format F10.1).
- c) RAD      The distance in nautical miles from the storm center to the region of maximum winds (format F10.1).

- d) VRMAX Maximum observed windspeed in knots (format F10.1).
- e) PZRO Atmospheric pressure in millibars of the storm center (format F10.1).

(3) Card 3

NT2 - Number of cards (max. 150) containing on each TIMPOS, XPOS and YPOS (format I5).

(4) Card Group 4

NT2 cards with each card containing values of

- a) TIMPOS The time in hours at which the hurricane position is recorded (format F10.1).
- b) XPOS The x-coordinate in units of x,y space of the hurricane center (format F10.1).
- c) YPOS The y-coordinate in units of x,y space of the hurricane center (format F10.1).

(5) Card Group 5

GRID2 The fluid depth in fathoms relative to mean sea level along each column,  $i=1,2..IM$ , is read with a nested do-loop for  $j=1,2..JM$  (format 11F7.2). The program will zero those values for even  $i+j$  prior to computations. The depth data are positive numbers which the program converts to negative values (in meters) to be consistent with the coordinate system.

(6) Card Group 6

S The values of the dimensionless scale factor relating the  $(x,y)$  plane to the  $(\xi,\eta)$  plane are read along each column,  $i=1,2..IM$ , with a nested do-loop for  $j=1,2..JM$  (format 5E14.7).

(7) Card Group 7

DSDXI The values of the dimensionless scale factor,  $\mu$ , transforming  $\xi$  to  $S^*$  are read with one value per card for  $i=1,2..IM$  (format E14.7).

(8) Card Group 8

DTDET        Values of the dimensionless scale factor  $\nu$  transforming  $\eta$  to  $T^*$  are read with one value per card for  $j=1,2..JM$  (format 2X,E14.7).

(9) Card Group 9

HOBS1        The values of the observed hourly water level in feet at grid location ( $IH1, JH1$ ) are read with 19 values per card in format F4.1.

(10) Card Groups 10 through 14

HOBX $k$         The values of the observed hourly water level in feet at grid location ( $IHk, JHk$ ) are read sequentially with the same format as above.

(11) Card Group 15

XX, YY        The paired  $x,y$  coordinates (units in  $x,y$  space) of the computational grid points are read along each column,  $i=1,2..IM$ , with a nested do-loop for  $j=1,2..JM$  (format 10F7.2).

(12) Card Group 16

COSG, SING    The paired values of the cosine  $\theta$  and sine  $\theta$  where  $\theta$  is the angle between the  $\xi$  and  $x$  axis are read along each column,  $i=1,2..IM$ , with a nested do-loop for  $j=1,2..JM$  (format 10F8.5).

A computer printout of the sample program and data cards are given later in this section.

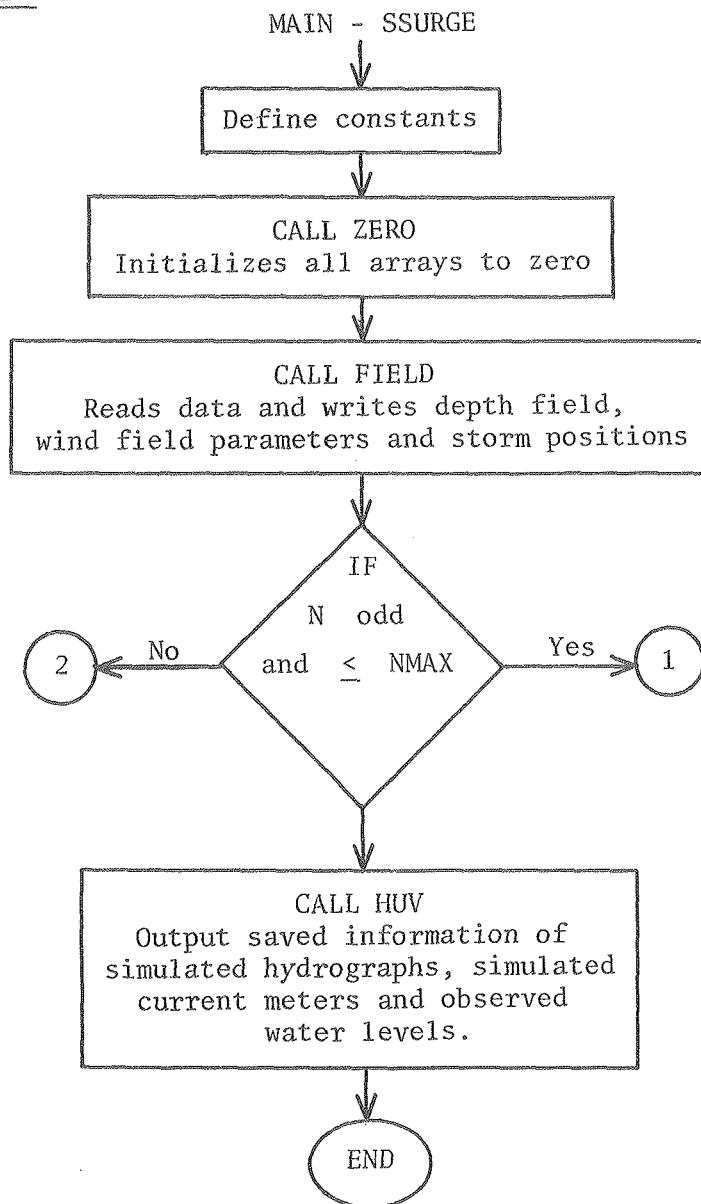
5. Mathematical Procedures and Program Limitations.

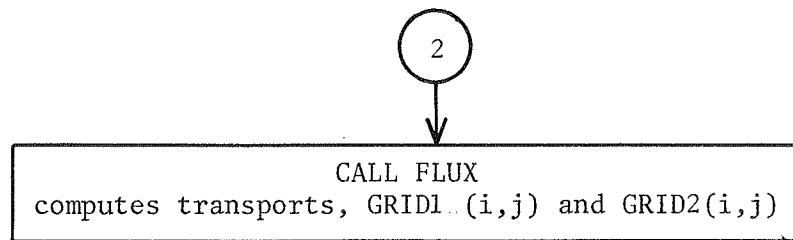
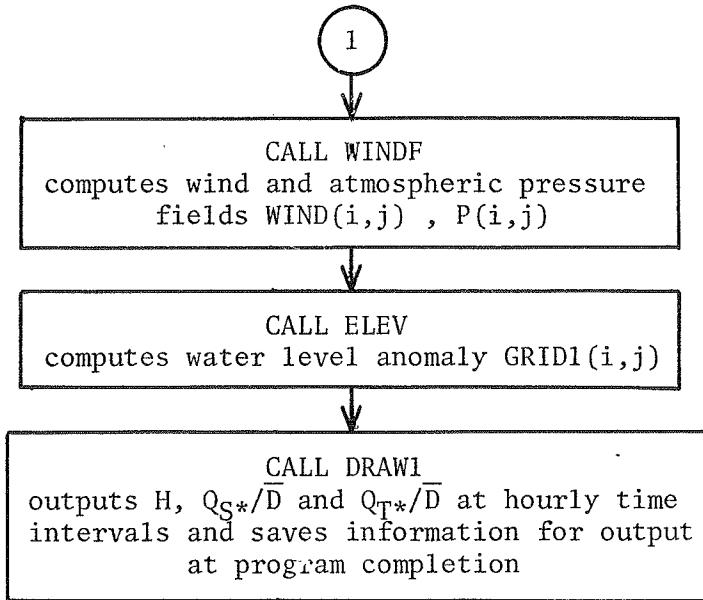
The storm surge equations, solutions, and algorithms are described in the text of Volume I of this report.

Basically, the model utilizes a centered difference, leapfrog analog of the vertically integrated, quasi-linear form of the long-wave equations. The algorithm treats the time dependency explicitly and employs a computing lattice in which the transports,  $Q_{S^*}$  and  $Q_{T^*}$ , are computed at the same location but are staggered in time and space with respect to the water

level anomaly. The program assumes that seabed scouring does not occur. No program error messages or consistency checks are incorporated in this version.

6. Flow Chart.





## 7. Glossary and Description of Terms.

### Arrays

GRID1	Dimensioned $IM \times JM$ . Contains $H$ values in meters at odd $i+j$ and $Q_{S^*}$ in meters squared per second at even $i+j$ .
GRID2	Dimensioned $IM \times JM$ . Contains fluid depth values, $D_o$ , in meters at odd $i+j$ and $Q_{T^*}$ in meters squared per second at even $i+j$ .
S	Dimensioned $IM \times JM$ . Contains <u>dimensionless</u> scale factor, $F$ , determined from the <u>conformal mapping</u> of $(x,y)$ space to $(\xi,\eta)$ space.
DSDXI	Dimensioned $IM$ . Contains the dimensionless scale factor, $\mu$ , transforming $\xi$ to $S^*$ .
DTDET	Dimensioned $JM$ . Contains dimensionless scale factor, $\nu$ , transforming $\eta$ to $T^*$ .
XX	Dimensioned $IM \times JM$ . Contains the x-coordinate in units of $(x,y)$ space of the grid point locations.
YY	Dimensioned $IM \times JM$ . Contains the y-coordinate in units of $(x,y)$ space of the grid point locations.
COSG	Dimensioned $IM \times JM$ . Contains cosine values of $\theta$ relating the orientation of the $\xi$ -axis to the x-axis at the computational grid points.
SING	Dimensioned $IM \times JM$ . Contains sine value of $\theta$ relating the orientation of the $\xi$ -axis to the x-axis at the computational grid points.
WIND	Dimensioned $IM + 1 \times JM$ . Contains values of the windspeed in meters per second. At a transport computational grid point $(i,j)$ , the $S^*$ -component is stored in WIND at $(i,j)$ and the $T^*$ -component is stored at $(i+1,j)$ .
P	Dimensioned $IM \times JM$ . Contains values of $H_B$ (the hydrostatic elevation in meters of the sea surface corresponding to the atmospheric pressure anomaly) and stored at odd $i+j$ .
$HOBS_k$ $k=1,2..6$	Each array is dimensioned 67. Contains the observed hourly water levels in meters at grid locations $(IH_k, JH_k)$ .

<u>HYDk</u> k=1,2..6	Each array is dimensioned 300. Contains values of the computed water level $H$ in meters at grid locations ( $IH_k$ , $JH_k$ ).
<u>UCMk</u> k=1,2..6	Each array is dimensioned 300. Contains values of the vertically averaged water velocity, $Q_{S^*}/\bar{D}$ , in meters per second at grid location ( $IT_k$ , $JT_k$ ).
<u>VCMk</u> k=1,2..6	Each array is dimensioned 300. Contains values of the vertically averaged water velocity, $Q_{T^*}/\bar{D}$ , in meters per second at grid location ( $IT_k$ , $JT_k$ ).
TIM	Dimensioned 50. Contains the time in hours at which the wind and atmospheric pressure field parameters (ROT, RAD, VRMAX and PZRO) are recorded.
ROT	Dimensioned 50. Contains the angle in degrees between the direction the storm is moving and the region of maximum winds.
RAD	Dimensioned 50. Contains the distance in nautical miles from the storm center to the region of maximum winds.
VRMAX	Dimensioned 50. Contains the maximum observed windspeed in knots.
PZRO	Dimensioned 50. Contains the atmospheric pressure in millibars of the storm center.
TIMPOS	Dimensioned 150. Contains the time in hours at which the storm position is recorded.
XPOS	Dimensioned 150. Contains the x-coordinate in units of (x,y) space of the storm center.
YPOS	Dimensioned 150. Contains the y-coordinate in units of (x,y) space of the storm center.

#### Constants

IM	Number of grid points in the $S^*$ direction.
JM	Number of grid points in the $T^*$ direction.
NMAX	Maximum number of time steps.

INC	Number of time steps between saving of surge results in HYD <sub>k</sub> , UCM <sub>k</sub> and VCM <sub>k</sub> . INC must be an even integer number.
DXI	Grid increment in meters in the S* direction.
DETA	Grid increment in units of meters in the T* direction. The units of DTDET·DETA (that is, $v \cdot \Delta T^*$ ) are in meters.
DELT	Time increment in seconds. DELT must be less than that required for numerical stability and, also, be an even integer multiple of 60 minutes.
GRAV	Acceleration due to gravity (=9.8 meters per second squared).
F	Dimensionless seabed drag coefficient (=0.0025).
IH <sub>k</sub> , JH <sub>k</sub> k=1, 2..6	Grid point location for saving the computed water level. The sum of the indexes must be odd. Index IH <sub>k</sub> cannot exceed IM. Index JH <sub>k</sub> cannot exceed JM.
IT <sub>k</sub> , JT <sub>k</sub> k=1, 2..6	Grid point location for saving the vertically averaged water velocities. The sum of the indexes must be even. Index IT <sub>k</sub> cannot exceed IM. Index JT <sub>k</sub> cannot exceed JM-1.
CORIO	Coriolis parameter ( $=6.70875 \times 10^{-5}$ second <sup>-1</sup> ) for latitude 27° 23.232' N.
PHI	Wind ingress angle in units of degrees.
PINF	Far field atmospheric pressure in millibars (=1016 millibars).

#### Comments

1. A symmetric analytical hurricane wind field representation as given by C. Jelesnianski (1965, A numerical calculation of storm tides induced by a tropical storm impinging on a continental shelf, *Mon. Wea. Rev.*, 94, 379-394) is employed in the surge model).
2. The wind stress coefficients, K<sub>1</sub> and K<sub>2</sub>, are  $1.1 \times 10^{-6}$  and  $2.5 \times 10^{-6}$  and are defined in SUBROUTINE FLUX.
3. Constants YRANGE, THIT, XHIT and YHIT are not used in this program version.

## 8. Input and Output.

Data statements in Program SSURGE establish the number of computational points, number of time steps, output interval, values of the grid and time steps, acceleration due to gravity, bottom stress coefficient, and locations of the simulated hydrographs and current meters, and the Coriolis parameter corresponding to latitude  $27^{\circ} 23.232' N$ .

Card input provides the number (NT1) of hurricane description sets, followed by NT1 cards giving the time (in hours after start of computations) and the three required storm parameters. These are succeeded by one card giving the number (NT2) of hurricane positions to be used, followed by NT2 cards providing the time and storm center positions in the original arbitrary Cartesian grid (see Volume I of this report). NT1 need not be equal to NT2, nor must the observed storm parameters and positions coincide in time.

For the Carla surge computations, the hurricane characteristics are stipulated at 6-hour intervals for the first 18 hours and at 3-hour intervals for the remainder of the 66-hour prototype time simulation. Note the radius to maximum winds are in *nautical miles*, the maximum winds are in *knots*, and the central pressures are in *millibars*. Due to the erratic movement of Hurricane Carla, the coordinates of the center of the storm are specified at hourly intervals, except for a single 6-hour interval spanning the end of the simulation. These coordinates are specified in *x,y* space (100 units = 219 kilometers).

The depth field, in *fathoms*, is introduced followed by the array of scale factors, *S*. The array of scale factors, DSDXI (an alias for  $\mu$ ) is read along the specified row and the scale factor array DTDET (an alias for  $\nu$ ) is specified along the chosen column.

The following six arrays are the observed hourly water levels (in feet) from Padre Island (HOBS1), Aransas Pass (HOBS2), Port O'Connor (HOBS 3), Pleasure Pier, Galveston (HOBS4), Mud Bayou (HOBS5), and Sabine Pass (HOBS6). These data are not necessary for any phase of the surge calculations and may be omitted.

The last two arrays input to SSURGE give the coordinates (XX,YY) of the grid points in units of *x,y* space, and the sines and cosines of theta at the computational points.

All input not in the MKS system of units is converted internally to the MKS system.

The storm parameters, hurricane center coordinates, and depth field are printed out immediately following input. At hourly time intervals, the water level anomalies and the depth averaged velocity component fields are printed. The six simulated hydrographs and current meters are printed out with their positions and the corresponding observed water levels at the completion of the surge simulation.

C  
C  
C

The following data are required for the Hurricane Carla surge simulation. The reader is referred to the documentation of Program GRID for definition of the scale factors, grid point coordinates, and the sines and cosines of theta.

C  
C  
C  
C  
C  
C

21 HURRICANE CARLA STORM PARAMETERS

0.	60.	23.	90.	949.
6.	35.	22.	95.	943.
12.	65.	21.	100.	940.
18.	65.	24.	100.	937.
21.	60.	23.	100.	937.
24.	80.	22.	100.	937.
27.	65.	25.	100.	937.
30.	80.	27.	100.	936.
33.	150.	19.	100.	936.
36.	15.	21.	100.	936.
39.	15.	20.	100.	937.
42.	65.	22.	100.	937.

45.	23.	26.	100.	937.
48.	65.	28.	100.	938.
51.	105.	28.	95.	938.
54.	160.	33.	90.	939.
57.	145.	62.	70.	940.
60.	158.	98.	55.	942.
63.	136.	142.	55.	943.
66.	140.	167.	50.	945.
69.	117.	200.	55.	947.

62 HURRICANE CARLA TRACK

0.	249.7	=82.1
1.	247.9	=75.3
2.	245.4	=68.5
3.	243.1	=61.0
4.	238.1	=53.1
5.	232.8	=50.0
6.	225.3	=44.3
7.	219.0	=37.7
8.	213.5	=32.0
9.	208.9	=26.5
10.	204.8	=21.2
11.	203.2	=12.6
12.	199.2	=8.1
13.	191.7	=6.2
14.	188.0	=4.1
15.	188.0	=1.0
16.	188.9	3.1
17.	189.9	9.0
18.	191.0	13.4

19.	192.1	18.5
20.	192.1	25.0
21.	191.9	31.0
22.	190.2	38.0
23.	188.5	43.8
24.	182.7	48.2
25.	181.1	51.3
26.	181.3	54.9
27.	184.0	61.2
28.	186.0	69.0
29.	186.8	74.0
30.	182.3	81.0
31.	177.5	88.7
32.	172.0	93.3
33.	162.4	92.4
34.	155.6	93.2
35.	152.7	95.7
36.	152.8	98.1
37.	154.3	103.1
38.	156.3	107.1
39.	158.9	112.8
40.	158.0	121.0
41.	152.2	131.3
42.	145.0	134.9
43.	141.6	130.2
44.	145.1	126.8
45.	148.8	129.4
46.	151.9	136.7
47.	155.5	146.3
48.	157.0	152.2
49.	157.4	158.2



10.70	8.50	6.00	1.50
108.00	60.00	47.00	36.00
10.50	8.00	6.00	1.50
108.00	61.00	47.00	37.50
10.00	8.00	6.00	1.50
102.00	61.00	36.00	36.00
9.50	8.00	6.00	1.50
108.00	60.00	40.00	35.00
9.00	7.70	5.50	1.30
110.00	55.00	35.00	23.00
11.00	7.70	5.50	1.30
102.00	36.00	31.00	27.50
11.00	7.70	5.50	1.30
110.00	35.00	31.00	25.00
9.50	8.00	6.00	1.30
115.00	34.50	29.00	25.00
9.00	8.00	6.00	1.30
100.00	34.00	28.00	24.00
8.50	7.50	5.50	1.30
100.00	34.00	27.00	24.00
8.70	7.00	5.50	1.30
100.00	35.00	27.00	24.00
8.50	7.00	5.50	1.30
100.00	36.00	27.00	24.00
8.70	7.50	5.50	1.20
102.00	34.00	26.50	22.00
8.00	7.30	5.50	1.20
100.00	35.00	27.00	23.00
9.00	7.30	5.00	1.20
100.00	36.00	27.00	22.00
7.00	6.20	4.00	1.00









1.4902367E+00 1.5151256E+00 1.5713631E+00 1.6068148E+00 1.6339124E+00  
1.6558009E+00 1.6745767E+00 1.6911773E+00 1.7058222E+00 1.7192673E+00  
1.7323044E+00 1.7470524E+00 1.7657062E+00 1.7907198E+00 1.8082151E+00  
1.5650425E+00 1.5873117E+00 1.6391338E+00 1.6747159E+00 1.7039928E+00  
1.7293187E+00 1.7524579E+00 1.7741182E+00 1.7940669E+00 1.8125184E+00  
1.8293711E+00 1.8468545E+00 1.8694602E+00 1.9021138E+00 1.9255364E+00  
1.6418245E+00 1.6616685E+00 1.7090353E+00 1.7442923E+00 1.7758022E+00  
1.8049900E+00 1.8332815E+00 1.8611928E+00 1.8879352E+00 1.9125255E+00  
1.9311228E+00 1.9380464E+00 1.9270075E+00 1.9004238E+00 1.8814047E+00  
1.7225043E+00 1.7381167E+00 1.7778070E+00 1.8121023E+00 1.8456544E+00  
1.8788385E+00 1.9128379E+00 1.9483185E+00 1.9846672E+00 2.0215143E+00  
2.0548605E+00 2.0774502E+00 2.0778233E+00 2.0607204E+00 2.0493568E+00  
1.7965964E+00 1.8079521E+00 1.8395907E+00 1.8723475E+00 1.9074805E+00  
1.9442498E+00 1.9836687E+00 2.0268065E+00 2.0738916E+00 2.1271386E+00  
2.1880780E+00 2.2634694E+00 2.3688770E+00 2.5493790E+00 2.6917623E+00  
1.8778135E+00 1.8827348E+00 1.9022305E+00 1.9324185E+00 1.9686607E+00  
2.0087526E+00 2.0534395E+00 2.1041572E+00 2.1618983E+00 2.2310803E+00  
2.3171132E+00 2.4346987E+00 2.6067113E+00 2.8674335E+00 3.0545618E+00  
1.9563065E+00 1.9531937E+00 1.9575452E+00 1.9844150E+00 2.0210031E+00  
2.0634128E+00 2.1119192E+00 2.1678973E+00 2.2321556E+00 2.3084771E+00  
2.3982261E+00 2.5018011E+00 2.6070034E+00 2.6950618E+00 2.7371290E+00  
2.0412583E+00 2.0277659E+00 2.0125789E+00 2.0352485E+00 2.0715286E+00  
2.1155270E+00 2.1668655E+00 2.2265745E+00 2.2948935E+00 2.3746489E+00  
2.4651689E+00 2.5640250E+00 2.6562304E+00 2.7198523E+00 2.7427186E+00  
2.1233956E+00 2.0988750E+00 2.0627685E+00 2.0808518E+00 2.1161562E+00  
2.1608188E+00 2.2137915E+00 2.2756525E+00 2.3459652E+00 2.4267147E+00  
2.5173933E+00 2.6229692E+00 2.7561489E+00 2.9517423E+00 3.0957615E+00  
2.2101971E+00 2.1725416E+00 2.1117769E+00 2.1248169E+00 2.1579439E+00  
2.2018436E+00 2.2550728E+00 2.3179536E+00 2.3894715E+00 2.4700442E+00  
2.5544966E+00 2.6348629E+00 2.6984390E+00 2.7643277E+00 2.8146395E+00  
2.2531337E+00 2.2091908E+00 2.1368146E+00 2.1474028E+00 2.1783285E+00

2.2205259E+00	2.2727737E+00	2.3356999E+00	2.4086962E+00	2.4927528E+00
2.5831397E+00	2.6723448E+00	2.7485091E+00	2.8323611E+00	2.8952547E+00
2.2515640E+00	2.2155313E+00	2.1567524E+00	2.1658727E+00	2.1919805E+00
2.2290327E+00	2.2769025E+00	2.3373294E+00	2.4115608E+00	2.5041475E+00
2.6168129E+00	2.7529640E+00	2.9098486E+00	3.0923688E+00	3.2103936E+00
2.1934112E+00	2.1798798E+00	2.1593699E+00	2.1679690E+00	2.1882922E+00
2.2183509E+00	2.2590232E+00	2.3127557E+00	2.3820815E+00	2.4738014E+00
2.5936655E+00	2.7491929E+00	2.9336425E+00	3.1266695E+00	3.2373314E+00
2.1029439E+00	2.1228970E+00	2.1560113E+00	2.1604552E+00	2.1688814E+00
2.1847413E+00	2.2095903E+00	2.2456951E+00	2.2956448E+00	2.3658014E+00
2.4632818E+00	2.6004555E+00	2.7866729E+00	3.0280126E+00	3.1872594E+00
2.0953010E+00	2.1231411E+00	2.1603608E+00	2.1434089E+00	2.1264820E+00
2.1154370E+00	2.1106526E+00	2.1124223E+00	2.1212619E+00	2.1383993E+00
2.1649236E+00	2.2025958E+00	2.2514537E+00	2.3093541E+00	2.3458089E+00
2.3079370E+00	2.2817307E+00	2.2051310E+00	2.1300498E+00	2.0667525E+00
2.0142219E+00	1.9685286E+00	1.9272917E+00	1.8896660E+00	1.8537017E+00
1.8184831E+00	1.7816093E+00	1.7421811E+00	1.7005746E+00	1.6753996E+00
2.8536853E+00	2.6981526E+00	2.3357893E+00	2.1416617E+00	2.0024239E+00
1.8916662E+00	1.7949727E+00	1.7057304E+00	1.6221904E+00	1.5414140E+00
1.4645743F+00	1.3918263F+00	1.3282498E+00	1.2792876E+00	1.2566270E+00
2.7518200E+00	1			
2.2581083E+00	2			
1.8063600E+00	3			
1.3889950E+00	4			
1.1300950E+00	5			
1.0143083E+00	6			
9.9518750E-01	7			
9.9441917E-01	8			
9.5426917E-01	9			
8.9961167E-01	10			
8.6152250E-01	11			

8.5889167E-01	12
9.0493250E-01	13
9.2801500E-01	14
1.0272283E+00	15
9.2585417E-01	16
7.4422667E-01	17
7.5768167E-01	18
7.2820500E-01	19
7.1056167E-01	20
7.0484833E-01	21
7.0268417E-01	22
6.9382417E-01	23
6.7679500E-01	24
6.4817583E-01	25
6.0436167E-01	26
5.6183500E-01	27
5.2465750E-01	28
4.6668333E-01	29
4.4389833E-01	30
4.9014500E-01	31
4.8859917E-01	32
3.7770917E-01	33
3.4322000E-01	34
4.3321667E-01	35
4.6491083E-01	36
4.1661167E-01	37
5.1132083E-01	38
5.2853250E-01	39
4.9022000E-01	40
5.2170083E-01	41
5.5028417E-01	42





















.94090 = .33868    .93360 = .35831    .92358 = .38340    .91067 = .41314    .89473 = .44661  
.87471 = .48465    .85035 = .52622    .82288 = .56822    .79417 = .60770    .77273 = .63473

APPENDIX A  
FORTRAN Listing of Program CONFORM

6

```
C PROGRAM CCNFORM. DETERMINES TRANSFROMATION COEFFS FOR CONFORMAL
C MAPPING OF INTERIOR REGION BOUNDED BY 2 CURVES INTO RECTANGLE.
C
C      DIMENSION X2(150) , Y2(150)
C      DIMENSION X2P(150) , Y2P(150)
C      DIMENSION COB(200) , COC(200)
C
C      COMMON/YB/ X2, Y2 , MQ
C      COMMON/YA/ X2P , Y2P , MQP
C      COMMON/COEF/           VARWT,JMAX1,IL
C      COMMON/FORIA/          COB , COC           , BZRO,XK,NMAX
C
C      READ CONTINUATION FLAG, NUMBER OF SHORELINE COORDS, NUMBER OF
C      SEAWARD BDRY COORDS, NUMBER OF COEFFS, MAX NUMBER OF ITERATIONS,
C      NUMBER OF INTEGRATION POINTS AND CONVERGENCE CRITERION
C      READ(5,1)IWANT, MQ, MQP, NMAX, JMAX1, IL, VARWT
C      1 FORMAT (6FI5, F5.3)
C
C
C      READ COORDS DELINEATUNG SHORELINE
C      DD 20 L=1,MQ
C      READ(5,11)           X2(L) , Y2(L)
C      11 FORMAT( 3X , F7.2 , 3X , F7.2 )
C      20 CONTINUE
C      READ COORDS DELINEATING SEAWARD BDRY
C      DO 30 J=1,MQP
C      READ (5,31) X2F(J) , Y2P(J)
C      30 CONTINUE
C      31 FORMAT( F7.2 , 3X , F7.2 )
C
```

```

PI= 3.1415927
XLAMDA= X2(MQ)
XK= PI/XLAMDA
C
      WRITE(6,9) NQ,NQP,XLAMDA,VARWT
9 FORMAT( 1H1 , /////
9//,20X,40HNUMBER OF COORDS DELINEATING SHORELINE= ,I3
9//,20X , 43HNUMBER OF COORDS DELINEATING SEWARD BDRY= ,I3
9      //,20X,32HHORIZONTAL EXTENT OF SHORELINE= ,F7.2
9//,20X,29HCONVERGENCE 'VARIANCE,' VARWT= ,F9.6
      WRITE(6,60) NMAX,JMAX1,IL
60   FORMAT( //,20X, 39HNUMBER OF TRANSFORMATION COEFFS ,NMAX= ,I3
9//,20X, 29HNUMBER OF ITERATIONS ,JMAX1= ,I3
9//,20X, 34HNUMBER OF INTEGRATION POINTS ,IL= ,I3
      WRITE(6,70)
70   FORMAT( 1H1 ,62X,16HSHORELINE COORDS ,/// )
      MQ13= MQ/3
      MQ3= MQ13*3
      IADD= MQ-MQ13
      DO 71 I=1,MQ13
      J= I+MQ13
      K= J+MQ13
      WRITE(6,72) I,I,X2(I),Y2(I), J,J,X2(J),Y2(J),K,K,X2(K),Y2(K)
71   FORMAT( /,3( 7X,3HX2(,I3,5H),Y2(,I3,3H)= ,F7.2,3X,F7.2 ) )
      IF( IADD .EQ. 0 ) GO TO 75
      IF( IADD .EG. 1 ) NUM1= MQ
      IF( IADD .EQ. 2 ) NUM1= MQ-1
      DO 73 I=NUM1,MQ
      73  WRITE(6,74) I,I,X2(I),Y2(I)
74   FORMAT( /, 89X,3HX2(,I3,5H),Y2(,I3,3H)= ,F7.2,3X,F7.2 )
75   CONTINUE

```

12

```
      WRITE(6,80)
80  FORMAT( 1H1      ,61X,19HSEAWARD BDRY COORDS      ,//   )
      MQP13= MQP/3
      MQP3= MQP13*3
      IADD= MQP=MQP3
      DO 81 I=1,MQP13
      J= I+MQP13
      K= J+MQP13
81  WRITE(6,82)I,I,X2P(I),Y2P(I),J,J,X2P(J),Y2P(J),K,K,X2P(K),Y2P(K)
82  FORMAT( /,3(5X,4HX2P(,I3,6H),Y2P(,I3,3H)= ,F7.2,3X,F7.2) )
      IF( IADD .EQ. 0 ) GO TO 85
      IF( IADD .EQ. 1 ) NUM1= MQP
      IF( IADD .EQ. 2 ) NUM1= MQP=1
      DO 83 I=NUM1,MQP
83  WRITE(6,84)I,I,X2P(I),Y2P(I)
84  FORMAT( /, 87X,4HX2P(,I3,6H),Y2P(,I3,3H)= ,F7.2,3X,F7.2 )
85  CONTINUE
C
C  DETERMINE COEFFS
C
      A= 0.0
      B= 0.0
      BZRO= 0.0
      WSX= 1.0
      WCX= 1.0
      WSY= 1.0
      WCY= 1.0
      DO 100 N=1,200
      COB(N)= 0.0
      COC(N)= 0.0
100  CONTINUE
      (WSX = 0.0)
      (WCX = 0.0)
```

```
      IF( IWANT .EQ. 1 ) GO TO 101
      GO TO 105
101  READ(5,102) B , BZRO
102  FORMAT( 2E14.7 )
      DO 103 N=1,NMAX
103  READ(5,102) COB(N) , COC(N)
      READ(5,104) WSX,WSY,WGX,WGY
104  FORMAT( 4E14.7 )
105  CONTINUE
C
C      A= -B
      CALL COEFFS( XLAMDA , A , B ,           WSX,WSY,WGX,WGY )
C
C      PUNCH 102, B,BZFO
C      DO 107 N=1,NMAX
C107  PUNCH 102, COB(N) , COC(N)
C      PUNCH 104, WSX,WSY, WGX, WGY
C
C
999  STOP
      END
      SUBROUTINE COEFFS( XLAMDA , A , B ,           WSX,WSY,WGX,WGY )
      DIMENSION COB(200) , COC(200)
      DIMENSION DLMB(200) , DUMC(200)
      DIMENSION X2P(150) , Y2(150)
      DIMENSION X2P(150) , Y2P(150)
      DIMENSION ARCYA(150) , ARCYB(150)
      DIMENSION SN(30,30) , CS(30,30)
      DIMENSION ATXA(400) , ATYA(400) , ATXB(400) , ATYB(400)
      DIMENSION U(400) , V(400) , Z(400)
      DIMENSION R(408) , S(408) , T(408) , W(400)
      DIMENSION X(408) , Y(408)
```

```
COMMON X,Y
COMMON/YB/ X2, Y2 , MQ
COMMON/YA/ X2P , Y2P , MQP
COMMON/JCHNR/ R
COMMON/JOHNS/ S
COMMON/JOHNT/ T
COMMON/LVW/ U , V , W
COMMON/AYA/ ARCYA
COMMON/AYE/ ARCYB
COMMON/COEF/           VARWT,JMAX1,IL
COMMON/FORIA/          COB , COC           , BZRO,XK,NMAX
C
SIGMA= -1.0
SIGMA1= -1.0
C DETERMINE THE NECESSARY PARAMETERS TO COMPUTE AN INTERPOLATORY
C SPLINE UNDER TENSION THROUGH A SEQUENCE OF FUNCTIONAL VALUES.
MQ1= MQ-1
MQP1= MQP-1
IYB= 0
DO 10 I=1,MQ1
IF( X2(I+1) .LT. X2(I) ) IYB= 1
10 CONTINUE
IF( IYB .EQ. 1 ) GO TO 15
CALL CUR1YB( SLP1 , SLPN , SIGMA )
INC= 11
XINC= INC-1
ARCYB( 1)= 0.0
DO 13 I=1,MQ1
K= I
II=I+1
DELX= X2( II )-X2( I )
```

```
    DELX= DELX/XINC
    SUM=0.0
    DO 12 J=1,INC
      XJ= J-1
      XDUM= X2(I) + XJ*DELX
      U(J)= CURDYB( XDUM , SIGMA , K )
      U(J)= U(J)*U(J) + 1.0
      U(J)= SQRT( U(J) )
12    SUM= SUM + U(J)
      SUM= SUM -0.5*U(1) - 0.5*U(INC)
13    ARCYB(II)= ARCYB(I) + SUM*DELX
      ARCB= ARCYB(MQ)
      GO TO 19
15    CONTINUE
      ARCYB(1)= 0.0
      DO 18 I=1,NQ1
        II= I+1
        DELX= X2(II)-X2(I)
        DELX= DELX*DELX
        DELY= Y2(II)-Y2(I)
        DELY= DELY*DELY
18    ARCYB(II)= ARCYB(I) + SQRT( DELX + DELY )
      ARCB= ARCYB(MQ)
19    CONTINUE
C
      CALL CUR2YB( SLP1 , SLPN , SIGMA)
      CALL CUR3YE( SLP1 , SLPN , SIGMA)
C
      IYA= 0
      DO 20 I=1,MQP1
        IF( X2P(I+1) .LT. X2P(I) ) IYA= 1
```

75

```
20  CONTINUE
    IF( IYA .EQ. 1 ) GO TO 25
    CALL CUR1YA( SLP1 , SLPN , SIGMA )
    INC= 11
    XINC= INC-1
    ARCYA(1)= 0.0
    DO 23 I=1,MQP1
    K= I
    II= I+1
    DELX= X2P(II)-X2P(I)
    DELX= DELX/XINC
    SUM=0.0
    DO 22 J=1,INC
    XJ= J-1
    XDUM= X2P(I) + XJ*DELX
    U(J)= CURDYA( XDUM , SIGMA1 , K )
    U(J)= U(J)*U(J) + 1.0
    U(J)= SQRT( U(J) )
22    SUM= SUM + U(J)
    SUM= SUM -0.5*U(1) - 0.5*U(INC)
23    ARCYA(II)= ARCYA(I) + SUM*DELX
    ARCA= ARCYA(MQP)
    GO TO 29
25  CONTINUE
    ARCYA(1)= 0.0
    DO 28 I=1,MQP1
    II= I+1
    DELX= X2P(II) - X2P(I)
    DELX= DELX*DELX
    DELY= Y2P(II) - Y2P(I)
    DELY= DELY*DELY
```

```
28 ARCYA(II)= ARCYA(I) + SQRT( DELX + DELY )
ARCA= ARCYA(MGF)
29 CONTINUE
C
CALL CUR2YA( SLP1 , SLPN , SIGMA1 )
CALL CUR3YA( SLP1 , SLPN , SIGMA1 )
C
WRITE(6,30)
30 FORMAT( 1H1, //,45X,40HARCLENGTH OF SEAWARD BDRY AND COASTLINE
9//,10X,25HARCLENGTH OF SEAWARD BDRY,8X,3H X2P,9X,3HY2P
915X,22HARCLENGTH OF COASTLINE,7X,2HX2,10X,2HY2P
DO 31 IT=1,MQP
IK= IT
ALNA= ARCYA(IT)
ATXA(IT)= CUR4YA( ALNA , SIGMA1 , IK )
31 ATYA(IT)= CUR5YA( ALNA , SIGMA1 , IK )
DO 32 IT=1,MC
IK= IT
ALNB= ARCYE(IT)
ATXB(IT)= CUR4YE( ALNB , SIGMA , IK )
32 ATYB(IT)= CUREYB( ALNB , SIGMA , IK )
IF( MQ .GE. MQP ) NUM2= MQP
IF( MQ .LE. MGP ) NUM2= MC
DO 33 I=1,NUM2
33 WRITE(6,34) ARCYA(I),ATXA(I),ATYA(I), ARCYB(I),ATXB(I),ATYB(I)
34 FORMAT(25X,F8.2,7X,F7.2,5X,F7.2,25X,F8.2,7X,F7.2,5X,F7.2
IF( MQ .EQ. MQP ) GO TO 41
IF( MQ .GT. MGF ) GO TO 35
IF( MQ .LT. MGP ) GO TO 38
35 NUM1= MQP+1
DO 36 I=NUM1,MC
```

36 WRITE(6,37) ARCYB(I),ATXB(I),ATYB(I) ←  
37 FORMAT( 84X,F8.2,7X,F7.2,5X,F7.2 )

38 GO TO 41

38 .NUM1= MG+1

39 DO 39 I=NUM1,MQP

39 WRITE(6,40) ARCYA(I),ATXA(I),ATYA(I) ←

40 FORMAT( 25X,F8.2,7X,F7.2,5X,F7.2 )

41 CONTINUE

C

C

IL12= IL/2

IL2= 2\*IL12

IADD= 0

IF( IL2 .NE. IL ) IADD= 1

IL1= IL12 + IADD

XIL= IL-1

DELXI= XLAMDA/XIL

FT= DELXI/XLAMDA

DO 45 IT=1,IL

XIT= IT-1

45 Z(IT)= XIT\*DELXI

NUM1= 1

46 NUM2= NMAX

DO 50 N=NUM1,NUM2

XN= N

XNK= XN\*XK

DO 50 IT=1,IL1

XNKKI= XNK\*Z(IT)

SN(N,IT)= SIN( XNKKI )

50 CS(N,IT)= COS( XNKKI )

C

78

```
ICOUNT= 1
55 CONTINUE
IF( ICOUNT .EQ. 1 ) GO TO 70
GO TO 79
70 IF( ABS(B) .LT. 0.001 ) GO TO 71
GO TO 76
71 YSUMA= 0.0
YSUMB= 0.0
DO 74 IT=1,IL
IK= IT
ALNA= ARCA*Z(IT)/XLAMDA
ALNB= ARCB*Z(IT)/XLAMDA
ATXA(IT)= CUR4YA( ALNA , SIGMA1 , IK )
ATYA(IT)= CUR5YA( ALNA , SIGMA1 , IK )
ATXB(IT)= CUR4YB( ALNB , SIGMA , IK )
ATYB(IT)= CUR5YE( ALNB , SIGMA , IK )
ATXA(IT)= ATXA(IT)-Z(IT)
ATXB(IT)= ATXB(IT)-Z(IT)
YSUMA= YSLNA + ATYA(IT)
74 YSUMB= YSUMB + ATYB(IT)
YSUMA= YSUMA = 0.5*ATYA(1) = 0.5*ATYA(IL)
YSUMB= YSLNB = 0.5*ATYB(1) = 0.5*ATYB(IL)
ADUM= YSUMA*FT*WSY
BDUM= YSUMB*FT*WCY
GO TO 81
C
76 CONTINUE
A= -B
ETA A= A
ETAB= B
SUMA= 0.0
```

79

```
SUMB= 0.0
DO 77 IT=1,IL
  XI= Z(IT)
  CALL SLFAC( XI, ETAA, DXDETA, DYDETA, SCFACA, SF )
  CALL SLFAC( XI, ETAB, DXDETA, DYDETA, SCFACB, SF )
  SUMA= SUMA + SCFACA
  SUMB= SUMB + SCFACB
  U(IT)= SCFACA
77  V(IT)= SCFACB
  SUMA= SUMA = 0.5*U(1) = 0.5*U(IL)
  SUMB= SUMB = 0.5*V(1) = 0.5*V(IL)
  ALNA= SUMA*DELXI
  ALNB= SUMB*DELXI
  XMODA= ARCA/ALNA
  XMODB= ARCB/ALNB
  SUMA= 0.0
  SUMB= 0.0
  ALNA= 0.0
  ALNB= 0.0
  DO 78 IT=1,IL
    IK= IT
    ATXA(IT)= CUR4YA( ALNA , SIGMA1 , IK )
    ATYA(IT)= CUR5YA( ALNA , SIGMA1 , IK )
    ATXB(IT)= CUR4YB( ALNB , SIGMA , IK )
    ATYB(IT)= CUR5YB( ALNB , SIGMA , IK )
    IF( IT .EQ. IL ) GO TO 78
    II= IT+1
    SUMA= SUMA + ( U(IT) + U(II) )*DELXI/2.0
    SUMB= SUMB + ( V(IT) + V(II) )*DELXI/2.0
    ALNA= SUMA*XMODA
    ALNB= SUMB*XMODB
```

08

```
78    CONTINUE
C
79    YSUMA= 0.0
    YSUMB= 0.0
    DO 80 IT=1, IL
    ATXA(IT)= ATXA(IT)-Z(IT)
    ATXB(IT)= ATXB(IT)-Z(IT)
    YSUMA= YSUMA + ATYA(IT)
80    YSUMB= YSUMB + ATYB(IT)
    YSUMA= YSUMA - 0.5*ATYA(1) - 0.5*ATYA(IL)
    YSUMB= YSUMB - 0.5*ATYB(1) - 0.5*ATYB(IL)
    ADUM= YSUMA*FT*WSY
    BDUM= YSUMB*FT*WCY
C
81    CONTINUE
    A11= WCY+WSY
    A12= WCY-WSY
    B1= BDUM-ADUM
    B2= BDUM+ADUM
    BOTTOM= A11*A11-A12*A12
    BTDUM= ( B1*A11-B2*A12 )/BOTTOM
    BZDUM= ( E2*A11-B1*A12 )/EOTTON
    BERR= BTDUM-B
    BZERR= BZDUM-BZRO
    B= BTDUM
    BZRO= BZDUM
    A= -B
C
    AVGWX= ( WSX+WCX )*0.5
    AVGWY= ( WSY+WCY )*0.5
    WBAR= ( WCX-WSX+WCY-WSY )*0.5
```

C

82     NUM2= NMAX  
     XKB= XK\*B  
     DO 89 N=1,NUM2  
     XASUMS= 0.0  
     XBSUMS= 0.0  
     YASUMC= 0.0  
     YBSUMC= 0.0  
     N12= N/2  
     N2= 2\*N12  
     DO 87 IT=1,IL  
      IF( IT .GT. IL1 ) GO TO 83  
      GO TO 85

83     IIT= IL+IT+1  
     SINN= SN(N,IIT)  
     COSS= CS(N,IIT)  
     IF( N .EQ. N2 ) GO TO 84  
     R(IT)= ATXA(IT)\*SINN  
     S(IT)= ATXB(IT)\*SINN  
     T(IT)= -ATYA(IT)\*COSS  
     W(IT)= -ATYE(IT)\*COSS  
     GO TO 86

84     R(IT)= -ATXA(IT)\*SINN  
     S(IT)= -ATXB(IT)\*SINN  
     T(IT)= ATYA(IT)\*COSS  
     W(IT)= ATYE(IT)\*COSS  
     GO TO 86

85     SINN= SN(N,IT)  
     COSS= CS(N,IT)  
     R(IT)= ATXA(IT)\*SINN  
     S(IT)= ATXB(IT)\*SINN

82

```
T( IT ) = A TYA( IT ) * COSS
W( IT ) = A TYB( IT ) * COSS
86    CONTINUE
XASUMS= XASUMS + R( IT )
XBSUMS= XESUMS + S( IT )
YASUMC= YASUMC + T( IT )
YBSUMC= YBSUMC + W( IT )
87    CONTINUE
XASUMS= XASUMS = 0.5*R( 1 ) = 0.5*R( IL )
XBSUMS= XBSUMS = 0.5*S( 1 ) = 0.5*S( IL )
YASUMC= YASUMC = 0.5*T( 1 ) = 0.5*T( IL )
YBSUMC= YBSUMC = 0.5*W( 1 ) = 0.5*W( IL )
XASUMS= XASUMS*FT*WSX
XBSUMS= XBSUMS*FT*WCX
YASUMC= YASUMC*FT*WSY
YBSUMC= YBSUMC*FT*WCY
XN= N
XNKB= XN*XKB
SHB= SINH( XNKB )
CHB= COSH( XNKB )
TB= SHB/CHB
SM= TE*TE
TM= SM/SHB
A11= SM*AVGWX + AVGWY
A12= TE*WEAR
A22= SM*AVGWY + AVGWX
B1= TM*(-XASUMS+XBSUMS) + (+YASUMC+YBSUMC)/CHB
B2= TM*(-YASUMC+YBSUMC) + (+XASUMS+XBSUMS)/CHE
BOTTOM= A11*A22 - A12*A12
DUMB(N)= ( B1*A22-B2*A12 )/BOTTOM
DUMC(N)= ( B2*A11-B1*A12 )/BOTTOM
```

```
89      CONTINUE
C
     EB1= DUME(1)=COB(1)
     EB2= DUMB(2)=COB(2)
     EC1= DUMC(1)=CCC(1)
     EC2= DUNC(2)=CCC(2)
C
     WRITE(6,250) ICCUNT , NMAX , IL
250   FORMAT( 1H1,20X,18H ITERATION NUMBER= ,I3,10X ,6HNMAX= ,I2,10X
     912HNUM POINTS= ,I4 )
C
     WRITE(6,260) A, E, BERR
260   FORMAT( 49X,7HALPHA= ,F11.6
     9/      ,50X,6HBETA= ,F11.6,3X,8HCHANGE= ,E10.3 )
C
     WRITE(6,261) BZRC , BZERR
261   FORMAT( 48X,8HB=ZERO= ,F11.6,3X,8HCHANGE= ,E10.3 ,/
C
     NUMB= 0
     NUMC= 0
     XKB= XK*B
     DO 270 N=1,NMAX
     XN= N
     XNKB= XN*XKB
     PRB= DUMB(N)*COB(N)
     PRC= DUMC(N)*CCC(N)
     IF( PRB .LT. 0.0 ) NUMB= NUMB + 1
     IF( PRC .LT. 0.0 ) NUMC= NUMC + 1
     ERRB= DUME(N)=COB(N)
     ERRC= DUNC(N)=CCC(N)
     COB(N)= DUMB(N)
```

```

COC(N)= DUMC(N)
AMPB= ABS( COB(N)*COSH(XNKB) )
AMPC= ABS( CCC(N)*SINH(XNKB) )
270 WRITE(6,271) N,COB(N),ERRB ,AMPB ,N,CCC(N),ERRC ,AMPC
271 FORMAT( 7X,2HB(, I3,3H)= ,E14.7,3X,E10.3,3X,E10.3 , 27X
         92HC(, I3,3H)= ,E14.7,3X,E10.3,3X,E10.3 )
C
C          WRITE(6,272) NUMB , NUMC
272 FORMAT(/,7X,34HNUM OF COEFFS THAT CHANGED SIGN= ,I3
         937X      ,34HNUM OF COEFFS THAT CHANGED SIGN= ,I3 ,// )
C
C
C DETERMINE THE VARIANCE BETWEEN THE PREDICTED COASTLINE AND
C SEAWARD BOUNDARY AND THAT SPECIFIED.
ETAA= A
ETAB= B
SUMA= 0.0
SUMB= 0.0
SUMEXA= 0.0
SUMEYA= 0.0
SUMEXB= 0.0
SUMEYB= 0.0
DO 277 IT=1,IL
XI= Z(IT)
R(IT)= XTRAN( XI , ETAA )
S(IT)= YTRAN( XI , ETAA )
T(IT)= XTRAN( XI , ETAB )
W(IT)= YTRAN( XI , ETAB )
CALL SLFAC( XI,ETAA,DXDETA,DYDETA,SCFACA,SF )
CALL SLFAC( XI,ETAB,DXDETA,DYDETA,SCFACB,SF )
SUMA= SUMA + SCFACA

```

```

SUMB= SUME + SCFACB
SUMXA= (ABS(ATXA(IT)+Z(IT)-R(IT)))**2
SUMYA= (ABS(ATYA(IT)-S(IT)))**2
SUMXB= (ABS(ATXB(IT)+Z(IT)-T(IT)))**2
SUMYB= (ABS(ATYB(IT)-W(IT)))**2
SUMEXA= SUMEXA + SUMXA
SUMEYA= SUMEYA + SUMYA
SUMEXB= SUMEXB + SUMXB
SUMEYB= SUMEYB + SUMYB
U(IT)= SCFACA
277 V(IT)= SCFACB
SUMEXA= SUMEXA - 0.5*SUMXA = 0.5*( ABS(ATXA(1)+Z(1)-R(1)) )**2
SUMEYA= SUMEYA - 0.5*SUMYA = 0.5*( ABS(ATYA(1)-S(1)) )**2
SUMEXB= SUMEXB - 0.5*SUMXB = 0.5*( ABS(ATXB(1)+Z(1)-T(1)) )**2
SUMEYB= SUMEYB - 0.5*SUMYB = 0.5*( ABS(ATYB(1)-W(1)) )**2
EFXA= SUMEXA*FT*WSX
EFYA= SUMEYA*FT*WSY
EFXB= SUMEXB*FT*WCX
EFYB= SUMEYB*FT*WCY
EF= EFXA+EFYA+EFXB+EFYB
EBAR= EF/4.0
WSY= EFYA/EBAR
WCY= EFYE/EBAR
WSX= EFXA/EBAR
WCX= EFXB/EBAR
SUMA= SUMA - 0.5*U(1) = 0.5*U(IL)
SUMB= SUMB - 0.5*V(1) = 0.5*V(IL)
ALNA= SUMA*DELEXI
ALNB= SUMB*DELEXI
XMODA= ARCA/ALNA
XMODB= ARCE/ALNB

```

(Remove)  
(Remove)

```

C
SUM A= 0.0
SUMB= 0.0
ALNA= 0.0
ALNB= 0.0
VARXA= 0.0
VARYA= 0.0
VARXB= 0.0
VARYB= 0.0
DO 278 IT=1, IL
IK= IT
X( IT )= ALNA
Y( IT )= ALNB
ATXA( IT )= CUR4YA( ALNA , SIGMA1 , IK )
ATYA( IT )= CUR5YA( ALNA , SIGMA1 , IK )
ATXB( IT )= CUR4YB( ALNB , SIGMA , IK )
ATYB( IT )= CUR5YB( ALNB , SIGMA , IK )
VARXA= VARXA + ABS( ATXA( IT )-R( IT ))**2
VARYA= VARYA + ABS( ATYA( IT )-S( IT ))**2
VARXB= VARXB + ABS( ATXB( IT )-T( IT ))**2
VARYB= VARYB + ABS( ATYB( IT )-W( IT ))**2
IF( IT .EQ. IL ) GO TO 278
II= IT+1
SUMA= SUMA + ( U( IT ) + U( II ) )*DELXI/2.0
SUMB= SUMB + ( V( IT ) + V( II ) )*DELXI/2.0
ALNA= SUMA*XMCDA
ALNB= SUMB*XMCDB
278 CONTINUE
C
VARXA= VARXA/XIL
VARYA= VARYA/XIL

```

87

```

VARXB= VARXB/XIL
VARYB= VARYB/XIL
VAR=      ( VARXA+VARYA+VARXB+VARYB )/4.0
VARA=      0.5*( VARXA+VARYA )
VARB=      0.5*( VARXE+VARYB )

C
C
      WRITE(6,280) XMCDE,XMODA, VARXB,EFXB,           VARYB,VARB,EFYB      ←
9VARXA,EFXA,          VARYA,VARA,EFYA, VAR , EF
280  FORMAT( 35X,5SHRATIO OF COASTLINE ARCLENGTH TO *TRANSFORM* GEN
9ERATED = ,F8.4
9/,32X,5SHRATIO OF SEAWARD BDRY ARCLENGTH TO *TRANSFCRM* GENERATED
9= ,F8.4 ,//, 66X,8H VARIANCE ,26X,7H ER FNCT
9//,7X,48H TRANSFCRM COASTLINE FRCM THAT SPECIFIED USING X ,E14.7
920X,E14.7
9/ ,7X,48H TRANSFCRM COASTLINE FROM THAT SPECIFIED USING Y ,E14.7
92X,E14.7,4X,E14.7
9/,4X,51H TRANSFORM SEAWARD BDRY FROM THAT SPECIFIED USING X ,E14.7,
920X,E14.7
9/,4X,51H TRANSFORM SEAWARD BDRY FRCM THAT SPECIFIED USING Y ,E14.7,
92X,E14.7,4X,E14.7
9/ ,71X,E14.7,20X,E14.7
      WRITE(6,281)                                     ←
281  FORMAT(//      ,15X,42H SEAWARD BDRY COORDS , O==XI==LAMDA , ETA=A,
923X           ,39H COASTLINE COORDS , O==XI==LAMDA , ETA=B
9/,8X,9HX(XI,ETA),2X,7HX=GIVEN,4X,9HY(XI,ETA),2X,7HY=GIVEN,4X
98HARC=TRAN,4X,1HI
98X ,9HX(XI,ETA),2X,7HX=GIVEN,4X,9HY(XI,ETA),2X,7HY=GIVEN,4X
98HARC=TRAN
C
      DO 285 I=1,IL

```

```
285  WRITE(6,286) I,R(I),ATXA(I),S(I),ATYA(I),X(I)
      9I,T(I),ATXB(I),W(I),ATYB(I),Y(I)
286  FORMAT( 4X,I3,2X,F7.2,3X,F7.2,5X,F7.2,3X,F7.2,4X,F8.2,4X,1HI
      94X           ,I3,2X,F7.2,3X,F7.2,5X,F7.2,3X,F7.2,4X,F8.2
)
C
C
      ICOUNT= ICOUNT + 1
      IF( VAR .LE. VARWT ) GO TO 375
300  IF( ICOUNT .LE. JMAX1 ) GO TO 55
375  RETURN
      END
      FUNCTION XTRAN( XI , ETA )
      DIMENSION COB(200) , COC(200)
      COMMON/FORIA/          COE , COC
                           , EZRO,XK,NMAX
      XTRAN= 0.0
      XK= XK*ETA
      XKXI= XK*XI
      DO 10 N=1,NMAX
      XN= N
      XNKA= XN*XKA
      XNKXI= XN*XKXI
      CSH= COSH(XNKA)
      SNH= SINH(XNKA)
      XTRAN=XTRAN+(COC(N)*CSH           +COB(N)*SNH           )*SIN(XNKXI)
10    CONTINUE
      XTRAN= XTRAN + XI
      RETURN
      END
      FUNCTION YTRAN( XI , ETA )
      DIMENSION COB(200) , COC(200)
      COMMON/FORIA/          COB , COC
                           , BZRO,XK,NMAX
```

```
YTRAN= 0.0
XKA= XK*ETA
XKXI= XK*XI
DO 10 N=1,NMAX
XN= N
XNKA= XN*XKA
XNKXI= XN*XKXI
CSH= COSH(XNKA)
SNH= SINH(XNKA)
YTRAN=YTRAN+(COB(N)*CSH      +COC(N)*SNH      )*COS(XNKXI)
10 CONTINUE
YTRAN= YTRAN + ETA + BZRO
RETURN
END
SUBROUTINE SLFAC( XI , ETA , DXDETA , DYDETA , SCFAC , SF )
DIMENSION CCE(200) , COC(200)
COMMON/FORIA/      CCB , COC           , BZRC,XK,NMAX
XKA= XK*ETA
XKXI= XK*XI
DXDETA= 0.0
DYDETA= 0.0
DO 10 N=1,NMAX
XN= N
XNKA= XN*XKA
XNKXI= XN*XKXI
CSH= COSH(XNKA)
SNH= SINH(XNKA)
XDETA= XN*XK*(COB(N)*CSH      +COC(N)*SNH      )*SIN(XNKXI)
YDETA= XN*XK*(COB(N)*SNH      +COC(N)*CSH      )*COS(XNKXI)
DXDETA= DXDETA + XDETA
DYDETA= DYDETA + YDETA
```

90

```
10  CONTINUE
    DYDETA= DYDETA + 1.0
    SF= (ABS(DXDETA))**2 + (AES(DYDETA))**2
    SCFAC= SQRT( SF )
    RETURN
    END
    SUBROUTINE CUR1YB( SLF1 , SLPN , SIGMA )
    DIMENSION X2(150) , Y2(150)
    DIMENSION A(150) , S(408)
    COMMON//JOHNA/ A
    COMMON//JOHNS/ S
    COMMON//YB/ X2, Y2 , MQ
C   FITS SPLINE== Y2 AS A FUNCTION OF X2
    N= MQ
    NM1 = N-1
    NP1 = N+1
    DELX1 = X2(2)-X2(1)
    DX1 = ((Y2(2)-Y2(1))/DELX1
    IF (SIGMA.LT.0.) GO TO 5
    SLPP1 = SLF1
    SLPPN = SLPN
1   SIGMAP = AES(SIGMA)*(N-1)/(X2(N)-X2(1))
    DELS = SIGMAP*DELX1
    EXP5 = EXP(DELS)
    SINHS = .5*(EXP5-1./EXP5)
    SINHIN = 1./(DELX1*SINHS)
    DIAG1 = SINHIN*(DELS*.5*(EXP5+1./EXP5)-SINHS)
    DIAGIN = 1./DIAG1
    A(1) = DIAGIN*(DX1-SLPP1)
    SPDIAG = SINHIN*(SINHS-DELS)
    S(1) = DIAGIN*SPDIAG
```

```
IF (N.EQ.2) GO TO 3
DO 2 I = 2,NM1
    DELX2 = X2(I+1)-X2(I)
    DX2= (Y2(I+1)-Y2(I))/DELX2
    DELS = SIGMAP*DELX2
    EXP5 = EXP(DELS)
    SINHS = .5*(EXP5-1./EXP5)
    SINHIN = 1./(DELX2*SINHS)
    DIAG2 = SINHIN*(DELS*(.5*(EXP5+1./EXP5))-SINHS)
    DIAGIN = 1./(DIAG1+DIAG2-SPDIAG*S(I-1))
    A(I) = DIAGIN*(DX2-SPDIAG*A(I-1))
    SPDIAG = SINHIN*(SINHS-DELS)
    S(I) = DIAGIN*SPDIAG
    DX1 = DX2
2   DIAG1 = DIAG2
3   DIAGIN = 1./(DIAG1-SPDIAG*S(NM1))
    A(N) = DIAGIN*(SLPPN=DX2-SPDIAG*A(NM1))
    DO 4 I = 2,N
        IBAK = NP1-I
4   A(IBAK) = A(IBAK)-S(IBAK)*A(IBAK+1)
    RETURN
5 IF (N.EQ.2) GO TO 6
    DELX2 = X2(3)-X2(2)
    DELX12 = X2(3)-X2(1)
    C1 = -(DELX12+DFLX1)/DELX12/DELX1
    C2 = DELX12/DELX1/DELX2
    C3 = -DELX1/DELX12/DELX2
    SLPP1 = C1*Y2(1) + C2*Y2(2) + C3*Y2(3)
    DELN = X2(N)-X2(NM1)
    DELNM1 = X2(NM1)-X2(N-2)
    DELNN = X2(N)-X2(N-2)
```

```

C1 = (DELNN+DELN)/DELNN/DELN
C2 = -DELNN/DELN/DELNN1
C3 = DELN/DELNN/DELNN1
SLPPN = C3*Y2(N-2) + C2*Y2(NM1) + C1*Y2(N)
GO TO 1
6 A(1) = 0.
A(2) = 0.
RETURN
END
FUNCTION CURVYB( T , SIGMA , IT )
DIMENSION X2(150) , Y2(150)
DIMENSION A(150)
COMMON/JOHNA/ A
COMMON/YB/ X2, Y2 , MG
C THIS FUNCTION INTERPOLATES THE COASTLINE AT A GIVEN VALUE FOR X.
C THE VALUE RETURNED IN CURVYB IS THE VALUE OF Y AT X.
C SUBROUTINE CUR1YB MUST BE CALLED EARLIER.
      N= MG
      S = X2(N)-X2(1)
      SIGMAP = ABS(SIGMA)*(N-1)/S
      IF (IT.EQ.1) I1 = 2
1 DO 2 I = I1,N
      IF( X2(I)=T ) 2,2,3
2 CONTINUE
      I = N
3 IF( X2(I-1) .LE. T .OR. T .LE. X2(1) ) GO TO 4
      I1 = 2
      GO TO 1
4 DEL1 = T-X2(I-1)
      DEL2 = X2(I)-T
      DELS = X2(I)-X2(I-1)

```

```
EXPS1 = EXP(SIGMA*DEL 1)
SINHD1 = .5*(EXPS1-1./EXP S1)
EXPS = EXP(SIGMAP*DEL 2)
SINHD2 = .5*(EXPS-1./EXPS)
EXPS = EXPS1*EXPS
SINHS = .5*(EXPS-1./EXPS)
CURVYB= ( A(I)*SINHD1+ A(I-1)*SINHD2)/SINHS+((Y2(I)-A(I))*DEL 1 +
9(Y2(I-1) - A(I-1))*DEL 2 )/DELS
I1 = I
RETURN
END
FUNCTION CURDYB( T , SIGMA , IT )
DIMENSION X2(150) , Y2(150)
DIMENSION A(150)
COMMON/JOHNA/ A
COMMON/YB/ X2, Y2 , MQ
C THIS FUNCTION DIFFERENTIATES THE COASTLINE AT A GIVEN VALUE
C FOR X. THE VALUE RETURNED IN CURDYB IS THE VALUE OF DY/DX AT X.
C SUBROUTINE CUR1YE MUST BE CALLED EARLIER.
N= MQ
S = X2(N)-X2(1)
SIGMAP = AES(SIGMA)*(N-1)/S
IF (IT.EQ.1) I1 = 2
1 DO 2 I = I1,N
   IF( X2(I)=T ) 2,2,3
2  CONTINUE
I = N
3  IF( X2(I-1) .LE. T .OR. T .LE. X2(1) ) GO TO 4
I1 = 2
GO TO 1
4 DEL1 = T-X2(I-1)
```

```

DEL2 = X2(I)-T
DELS = X2(I)-X2(I-1)
EXPS1 = EXP(SIGMAP*DELI)
COSHD1 = .5*(EXPS1+1./EXPS1)
EXPS = EXP(SIGMAP*DEL2)
COSHD2 = .5*(EXPS+1./EXPS)
EXPS = EXPS1*EXPS
SINHS = .5*(EXPS-1./EXPS)/SIGMAP
CURDYB= ( A(I)*COSHD1- A(I-1)*COSHD2)/SINHS+((Y2(I)-A(
1           I))-(Y2(I-1)-A(I-1)))/DELS
I1 = I
RETURN
END
SUBROUTINE CUR2YB( SLP1 , SLPN , SIGMA )
DIMENSION ARCYB(150)
DIMENSION X2(150) , Y2(150)
DIMENSION C(150) , S(408)
COMMON/JOHNC/ C
COMMON/JCHNS/ S
COMMON/AYB/ ARCYB
COMMON/YE/ X2, Y2 , MQ
C FITS SPLINE== X2 AS A FUNCTION OF ARCYB
N= MQ
NM1 = N-1
NP1 = N+1
DELX1 = ARCYB(2)-ARCYB(1)
DX1 = (X2(2)-X2(1))/DELX1
IF (SIGMA.LT.0.) GO TO 5
SLPP1 = SLP1
SLPPN = SLPN
1 SIGMAP = ABS(SIGMA)*(N-1)/(ARCYB(N)-ARCYB(1))

```

56

```
DELS = SIGMAP*DELX1
EXP S = EXP(DELS)
SINHS = .5*(EXP S=1. /EXP S)
SINHIN = 1. /(CFLX1*SINHS)
DIAG1 = SINHIN*(DELS*.5*(EXP S+1./EXP S)=SINHS)
DIAGIN = 1./DIAG1
C(1) = DIAGIN*(DX1=SLPP1)
SPDIAG = SINHIN*(SINHS=DELS)
S(1) = DIAGIN*SPDIAG
IF (N.EQ.2) GO TO 3
DO 2 I = 2,NM1
  DELX2 = ARCYB(I+1)-ARCYB(I)
  DX2= (X2(I+1)-X2(I))/DELX2
  DELS = SIGMAP*DELX2
  EXP S = EXP(DELS)
  SINHS = .5*(EXP S=1. /EXP S)
  SINHIN = 1. /(DELX2*SINHS)
  DIAG2 = SINHIN*(DELS*.5*(EXP S+1./EXP S)=SINHS)
  DIAGIN = 1./((DIAG1+DIAG2=SPDIAG*S(I-1))
    C(I) = DIAGIN*(DX2=DX1=SPDIAG*C(I-1))
  SPDIAG = SINHIN*(SINHS=DELS)
  S(I) = DIAGIN*SPDIAG
  DX1 = DX2
2  DIAG1 = DIAG2
3  DIAGIN = 1./((DIAG1=SPDIAG*S(NM1))
  C(N) = DIAGIN*(SLPPN=DX2=SPDIAG*C(NM1))
  DO 4 I = 2,N
    IBAK = NP1-I
4  C(IBAK) = C(IBAK)= S(IEAK)* C(IBAK+1)
  RETURN
5  IF (N.EQ.2) GO TO 6
```

```
DELX2 = ARCYB(3)-ARCYB(2)
DELX12 = ARCYE(3)-ARCYB(1)
C1 = -(DELX12+DELX1)/DELX12/DELX1
C2 = DELX12/DELX1/DELX2
C3 = -DELX1/DELX12/DELX2
SLPP1 = C1*X2(1) + C2*X2(2) + C3*X2(3)
DELN = ARCYB(N)-ARCYB(NM1)
DELNM1 = ARCYE(NM1)-ARCYE(N-2)
DELNN = ARCYB(N)-ARCYB(N-2)
C1 = (DELNN+DELN)/DELNN/DELN
C2 = -DELNN/DELN/DELNM1
C3 = DELN/DELNN/DELNM1
SLPPN = C3*X2(N-2) + C2*X2(NM1) + C1*X2(N)
GO TO 1
6  C(1) = 0.
    C(2) = 0.
RETURN
END
FUNCTION CUR4YB( T , SIGMA , IT )
DIMENSION ARCYB(150)
DIMENSION X2(150) , Y2(150)
DIMENSION C(150)
COMMON/JCHNC/ C
COMMON/AYE/ ARCYB
COMMON/YB/ X2, Y2 , MQ
C THIS FUNCTION INTERPOLATES THE COASTLINE AT A GIVEN VALUE FOR
C THE ARCLENGTH. THE VALUE RETURNED IN CUR4YB IS THE VALUE OF X
C AT ITS PARTICULAR ARCLENGTH. SUBROUTINE CUR2YE MUST BE CALLED BEFORE.
    N= MQ
    S = ARCYE(N)-ARCYE(1)
    SIGMAP = ABS(SIGMA)*(N-1)/S
```

```
      IF (IT.EQ.1) I1 = 2
1 DO 2 I = I1,N
     .IF( ARCYE(I)=T ) 2,2,3
2  CONTINUF
   I = N
3  IF( ARCYE(I-1) .LE. T .OR. T .LE. ARCYB(1) ) GO TO 4
   I1 = 2
   GO TO 1
4  DEL1 = T-ARCYB(I-1)
  DEL2 = ARCYB(I)-T
  DELS = ARCYB(I)-ARCYB(I-1)
  EXPS1 = EXP(SIGMAF*DEL1)
  SINHD1 = .5*(EXPS1-1./EXPS1)
  EXPS = EXP(SIGMAP*DEL2)
  SINHD2 = .5*(EXPS-1./EXPS)
  EXPS = EXPS1*EXPS
  SINHS = .5*(EXPS-1./EXPS)
  CUR4YB= ( C(I)*SINHD1+ C(I-1)*SINHD2)/SINHS+((X2(I)-C(I))*DEL1 +
9(X2(I-1)-C(I-1))*DEL2)/DELS
   I1 = I
   RETURN
END
SUBROUTINE CUR3YB( SLP1 , SLPN , SIGMA )
DIMENSION ARCYB(150)
DIMENSION X2(150) , Y2(150)
DIMENSION D(150) , S(408)
COMMON/JOHND/ D
COMMON/JOHNS/ S
COMMON/AYF/ ARCYB
COMMON/YB/ X2, Y2 , MQ
C FITS SPLINE== Y2 AS A FUNCTION OF ARCYB
```

```

N= MQ
NM1 = N-1
NP1 = N+1
DELX1 = ARCYB(2)-ARCYB(1)
DX1 = (Y2(2)-Y2(1))/DELX1
IF (SIGMA.LT.C.) GO TO 5
SLPP1 = SLP1
SLPPN = SLPN
1 SIGMAP = ABS(SIGMA)*(N-1)/(ARCYB(N)-ARCYB(1))
DELS = SIGMAP*DELX1
EXPS = EXP(DELS)
SINHS = .5*(EXPS-1./EXPS)
SINHIN = 1./(DELX1*SINHS)
DIAG1 = SINHIN*(DELS*.5*(EXPS+1./EXPS)-SINHS)
DIAGIN = 1./DIAG1
D(1) = DIAGIN*(DX1-SLPP1)
SPDIAG = SINHIN*(SINHS-DELS)
S(1) = DIAGIN*SPDIAG
IF (N.EQ.2) GC TC 3
DO 2 I = 2,NM1
    DELX2 = ARCYB(I+1)-ARCYB(I)
    DX2= (Y2(I+1)-Y2(I))/DELX2
    DELS = SIGMAP*DELX2
    EXPS = EXP(DELS)
    SINHS = .5*(EXPS-1./EXPS)
    SINHIN = 1./(DELX2*SINHS)
    DIAG2 = SINHIN*(DELS*.5*(EXPS+1./EXPS))-SINHS)
    DIAGIN = 1./(DIAG1+DIAG2-SPDIAG*S(I-1))
    D(I) = DIAGIN*(DX2-DX1-SPDIAG*D(I-1))
    SPDIAG = SINHIN*(SINHS-DELS)
    S(I) = DIAGIN*SPDIAG

```

```
      DX1 = DX2
2  DIAG1 = DIAG2
3  DIAGIN = 1./(DIAG1=SPDIAG* S(NM1))
   D(N) = DIAGIN*(SLPPN=DX2=SPDIAG* D(NM1 ))
   DO 4 I = 2,N
     IBAK = NF1=I
4  D(IBAK) = D(IBAK)= S(IBAK)* D(IEAK+1)
   RETURN
5  IF (N.EQ.2) GO TO 6
   DELX2 = ARCYB(3)=ARCYB(2)
   DELX12 = ARCYB(3)=ARCYB(1)
   C1 = -(DELX12+DELX1)/DELX12/DELX1
   C2 = DELX12/DELX1/DELX2
   C3 = -DELX1/DELX12/DELX2
   SLPP1 = C1*Y2(1) + C2*Y2(2) + C3*Y2(3)
   DELN = ARCYB(N)=ARCYB(NM1)
   DELNM1 = ARCYE(NM1)=ARCYB(N=2)
   DELNN = ARCYB(N)=ARCYE(N=2)
   C1 = (DELNN+DELN)/DELNN/DELN
   C2 = -DELNN/DELN/DELNM1
   C3 = DELN/DELNN/DELNM1
   SLPPN = C3*Y2(N=2) + C2*Y2(NM1) + C1*Y2(N)
   GO TO 1
6  D(1) = 0.
   D(2) = 0.
   RETURN
END
FUNCTION CUREYB( T , SIGMA , IT )
DIMENSION ARCYB(150)
DIMENSION X2(150) , Y2(150)
DIMENSION D(150)
```

100

```
COMMON/JCHND/ D
COMMON/AYE/ ARCYB
COMMON/YB/ X2, Y2 , MG
C THIS FUNCTION INTERPOLATES THE COASTLINE AT A GIVEN VALUE FOR
C THE ARCLENGTH. THE VALUE RETURNED IN CUR5YB IS THE VALUE OF Y
C AT ITS PARTICULAR ARCLENGTH. SUBROUTINE CUR3YE MUST BE CALLED BEFORE.
N= MQ
S = ARCYE(N)-ARCYB(1)
SIGMAP = ABS(SIGMA)*(N-1)/S
IF (IT.EQ.1) I1 = 2
1 DO 2 I = I1,N
   IF( ARCYB(I)=T ) 2,2,3
2  CONTINUE
I = N
3  IF( ARCYB(I-1) .LE. T .OR. T .LE. ARCYB(1) ) GO TO 4
   I1 = 2
   GO TO 1
4  DEL1 = T-ARCYB(I-1)
   DEL2 = ARCYB(I)-T
   DELS = ARCYB(I)-ARCYB(I-1)
   EXPS1 = EXP(SIGMAP*DEL1)
   SINHD1 = .5*(EXPS1-1./EXPS1)
   EXPS = EXP(SIGMAP*DEL2)
   SINHD2 = .5*(EXPS-1./EXPS)
   EXPS = EXPS1*EXPS
   SINHS = .5*(EXPS-1./EXPS)
   CUR5YB= ( D(I)*SINHD1+ D(I-1)*SINHD2)/SINHS+((Y2(I)-D(I))*DEL1 +
9(Y2(I-1) - D(I-1))*DEL2 )/DELS
   I1 = I
   RETURN
END
```

```
SUBROUTINE CUR1YA( SLP1 , SLPN , SIGMA )
DIMENSION B(150) , S(408)
DIMENSION X2P(150) , Y2P(150)
COMMON/JCHNE/ B
COMMON/JOHNS/ S
COMMON/YA/ X2P , Y2P , MQP
C FITS SPLINE=Y2P AS A FUNCTION OF X2P
      N= MQP
      NM1 = N-1
      NP1 = N+1
      DELX1 = X2P(2)-X2P(1)
      DX1 = (Y2P(2)-Y2P(1))/DELX1
      IF (SIGMA.LT.0.) GO TO 5
      SLPP1 = SLP1
      SLPPN = SLPN
      1 SIGMAP = ABS(SIGMA)*(N-1)/(X2P(N)-X2P(1))
      DELS = SIGMAP*DELX1
      EXP5 = EXP(DELS)
      SINHS = .5*(EXP5-1./EXP5)
      SINHIN = 1./(DELX1*SINHS)
      DIAG1 = SINHIN*(DELS*.5*(EXP5+1./EXP5)-SINHS)
      DIAGIN = 1./DIAG1
      B(1) = DIAGIN*(DX1-SLPP1)
      SPDIAG = SINHIN*(SINHS-DELS)
      S(1) = DIAGIN*SPDIAG
      IF (N.EQ.2) GO TO 3
      DO 2 I = 2,NM1
      DELX2 = X2P(I+1)-X2P(I)
      DX2 = (Y2P(I+1)-Y2P(I))/DELX2
      DELS = SIGMAP*DELX2
      EXP5 = EXP(DELS)
```

```
SINHS = .5*(EXPS-1./EXPS)
SINHIN = 1./(DELX2*SINHS)
DIAG2 = SINHIN*(DELS*(.5*(EXPS+1./EXPS))-SINHS)
DIAGIN = 1./(DIAG1+DIAG2-SPDIAG*S(I-1))
B(I) = DIAGIN*(DX2=DX1-SPDIAG*B(I-1))
SPDIAG = SINHIN*(SINHS-DELS)
S(I) = DIAGIN*SPDIAG
DX1 = DX2
2_ DIAG1 = DIAG2
3_ DIAGIN = 1./(DIAG1-SPDIAG*S(NM1))
B(N) = DIAGIN*(SLPPN=DX2-SPDIAG*B(NM1))
DO 4 I = 2,N
IBAK = NF1-I
4_ B(IBAK) = B(IBAK)- S(IBAK)* B(IBAK+1)
RETURN
5 IF (N.GC.TC.6
DELX2 = X2P(3)-X2P(2)
DELX12 = X2P(3)-X2P(1)
C1 = -(DELX12+DELX1)/DELX12/DELX1
C2 = DELX12/DELX1/DELX2
C3 = -DELX1/DELX12/DELX2
SLPP1 = C1*Y2P(1)+ C2*Y2P(2)+ C3*Y2P(3)
DELN = X2P(N)-X2P(NM1)
DELNM1 = X2P(NM1)-X2P(N-2)
DELNN = X2P(N)-X2P(N-2)
C1 = (DELNN+DELN)/DELNN/DELN
C2 = -DELNN/DELN/DELNM1
C3 = DELN/DEL NN/DEL NM1
SLPPN = C3*Y2P(N-2)+ C2*Y2P(NM1)+ C1*Y2P(N)
GO TO 1
6 B(1) = 0.
```

```
B(2) = 0.  
RETURN  
END  
FUNCTION CURVYA( T , SIGMA , IT )  
DIMENSION X2F(150) , Y2P(150)  
DIMENSION B(150)  
COMMON/JOHNP/ B  
COMMON/YA/ X2F , Y2P , MQP  
C THIS FUNCTION INTERPOLATES THE SEAWARD BDRY AT A GIVEN VALUE FOR X.  
C THE VALUE RETURNED IN CURVYA IS THE VALUE OF Y AT X.  
C SUBROUTINE CUR1YA MUST BE CALLED EARLIER.  
N= MQP  
S = X2P(N)-X2P(1)  
SIGMAP1= ABS(SIGMA)*(N-1)/S  
IF (IT.EQ.1) I1 = 2  
1 DO 2 I = I1,N  
IF( X2P(I)=T ) 2,2,3  
2 CONTINUE  
I = N  
3 IF( X2P(I-1).LE. T .OR. T .LE. X2P(1) ) GO TO 4  
I1 = 2  
GO TO 1  
4 DEL1 = T-X2P(I-1)  
DEL2 = X2P(I) - T  
DELS = X2P(I)-X2P(I-1)  
EXPS1 = EXP(SIGMA*DEL1)  
SINHD1 = .5*(EXPS1-1./EXPS1)  
EXPS = EXP(SIGMAP*DEL2)  
SINHD2 = .5*(EXPS-1./EXPS)  
EXPS = EXPS1*EXPS  
SINHS = .5*(EXPS-1./EXPS)
```

```
CURVYA = ( E(I)*SINHD1+ B(I-1)*SINHD2)/SINHS+((Y2P(I)-B(I))*DEL1 +  
9(Y2P(I=1) -E(I-1))*DEL2 )/DELS  
I1 = I  
RETURN  
END  
FUNCTION CURDYA( T , SIGMA , IT )  
DIMENSION X2P(150) , Y2P( 150)  
DIMENSION E(150)  
COMMON/JOHNB/ B  
COMMON/YA/ X2P , Y2P , MQP  
C THIS FUNCTION DIFFERENTIATES THE SEAWARD BDRY AT A GIVEN VALUE  
C FOR X. THE VALUE RETURNED IN CURDYA IS THE VALUE OF DY/DX AT X.  
C SUBROUTINE CURIYA MUST BE CALLED EARLIER.  
N= MQP  
S = X2P(N)-X2P(1)  
SIGMAP = ABS(SIGMA)*(N-1)/S  
IF (IT.EQ.1) I1 = 2  
1 DO 2 I = I1,N  
IF( X2P(I)=T ) 2,2,3  
2 CONTINUE  
I = N  
3 IF( X2P(I=1) .LE. T .OR. T .LE. X2P(1) ) GO TO 4  
I1 = 2  
GO TO 1  
4 DEL1 = T-X2P(I-1)  
DEL2 = X2P(I) - T  
DELS = X2P(I)-X2P(I-1)  
EXPS1 = EXP(SIGMAP*DEL1)  
COSHD1 = .5*(EXPS1+1./EXPS1)  
EXPS = EXP(SIGMAP*DEL2)  
COSHD2 = .5*(EXPS+1./EXPS)
```

50

```
EXPS = EXPS1*EXPS
SINHS = .5*(EXPS-1./EXPS)/SIGMAP
CURDY A= ( B(I)*COSHD1- B(I-1)*COSHD2)/SINHS+((Y2P(I)-B(
1      I))-(Y2P(I-1)-B(I-1)))/DELS
I1 = I
RETURN
END
SUBROUTINE CUR2YA( SLP1 , SLPN , SIGMA )
DIMENSION ARCYA(150)
DIMENSION E(150) , S(408)
DIMENSION X2P(150) , Y2P(150)
COMMON/JCHNE/ E
COMMON/JOHNS/ S
COMMON/AYA/ ARCYA
COMMON/YA/ X2P , Y2P , MGF
C   FITS SPLINE--X2P AS A FUNCTION OF ARCYA
N= MGP
NM1 = N-1
NP1 = N+1
DELX1 = ARCYA(2)-ARCYA(1)
DX1 = (X2P(2)-X2P(1))/DELX1
IF (SIGMA.LT.C.) GO TO 5
SLPP1 = SLF1
SLPPN = SLPN
1 SIGMAP = AES(SIGMA)*(N-1)/(ARCYA(N)-ARCYA(1))
DELS = SIGMAP*DELX1
EXPS = EXP(DELS)
SINHS = .5*(EXPS-1./EXPS)
SINHIN = 1./(DELX1*SINHS)
DIAG1 = SINHIN*(DELS*.5*(EXPS+1./EXPS)-SINHS)
DIAGIN = 1./DIAG1
```

```
E(1) = DIAGIN*(DX1=SLPP1)
SPDIAG = SINHIN*(SINHS=DELS)
S(1) = DIAGIN*SPDIAG
IF (N.EQ.2) GO TO 3
DO 2 I = 2,NM1
  DELX2 = ARCYA(I+1)=ARCYA(I)
  DX2= (X2P(I+1)-X2F(I))/DELX2
  DELS = SIGMAP*DELX2
  EXPS = EXP(DELS)
  SINHS = .5*(FXFS=1./EXPS)
  SINHIN = 1./(DELX2*SINHS)
  DIAG2 = SINHIN*(DELS*(.5*(EXPS+1./EXPS))-SINHS)
  DIAGIN = 1./(DIAG1+DIAG2=SPDIAG*S(I-1))
  E(I) = DIAGIN*(DX2=DX1=SPDIAG* E(I-1))
  SPDIAG = SINHIN*(SINHS=DELS)
  S(I) = DIAGIN*SPDIAG
  DX1 = DX2
2  DIAG1 = DIAG2
3  DIAGIN = 1./(DIAG1=SPDIAG*S(NM1))
  E(N) = DIAGIN*(SLPPN=DX2=SPDIAG* E(NM1))
  DO 4 I = 2,1
    IBAK = NP1=I
4  E(IBAK) = E(IBAK)= S(IBAK)* E(IBAK+1)
  RETURN
5  IF (N.EQ.2) GO TO 6
  DELX2 = ARCYA(3)=ARCYA(2)
  DELX12 = ARCYA(3)=ARCYA(1)
  C1 = -(DELX12+DELX1)/DELX12/DELX1
  C2 = DELX12/DELX1/DELX2
  C3 = -DELX1/DELX12/DELX2
  SLPP1 = C1*X2P(1)+ C2*X2P(2)+ C3*X2P(3)
```

```
DELN = ARCYA(N)=ARCYA(NM1)
DELNM1 = ARCYA(NM1)=ARCYA(N=2)
DELNN = ARCYA(N)=ARCYA(N=2)
C1 = .(DELNN+DELN)/DELNN/DELN
C2 = -DELNN/DELN/DELNM1
C3 = DELN/DELNN/DELNM1
SLPPN = C3*X2P(N=2)+ C2*X2P(NM1)+ C1*X2P(N)
GO TO 1
6 E(1) = 0.
E(2) = 0.
RETURN
END
FUNCTION CUR4YA( T , SIGMA , IT )
DIMENSION ARCYA(150)
DIMENSION X2P(150) , Y2P(150)
DIMENSION E(150)
COMMON/JCHNE/ E
COMMON/AYA/ ARCYA
COMMON/YA/ X2P , Y2P , MQP
C THIS FUNCTION INTERPOLATES THE SEAWARD ECRY AT A GIVEN VALUE FOR
C THE ARCLENGTH. THE VALUE RETURNED IN CUR4YA IS THE VALUE OF X
C AT ITS PARTICULAR ARCLENGTH. SUBROUTINE CUR2YA MUST BE CALLED BEFORE.
N= MQP
S = ARCYA(N)=ARCYA(1)
SIGMAP = ABS(SIGMA)*(N-1)/S
IF (IT.FQ.1) I1 = 2
1 DO 2 I = I1,N
    IF( ARCYA(I)=T ) 2,2,3
2 CONTINUE
I = N
3 IF( ARCYA(I=1) .LE. T .OR. T .LE. ARCYA(1) ) GO TO 4
```

```
    I1 = 2
    GO TO 1
4  DEL1 = T-ARCYA(I-1)
    DEL2 = ARCYA(I)-T
    DELS = ARCYA(I)-ARCYA(I-1)
    EXPS1 = EXP(SIGMAP*DEL1)
    SINHD1 = .5*(EXFS1-1./EXPS1)
    EXPS = EXP(SIGMAP*DEL2)
    SINHD2 = .5*(EXPS-1./EXPS)
    EXPS = EXPS1*EXPS
    SINHS = .5*(EXPS-1./EXPS)
    CUR4YA = ( E(I)*SINHD1+ E(I-1)*SINHD2)/SINHS+((X2P(I)-E(I))*DEL1 +
9(X2P(I-1)-E(I-1))*DEL2)/DELS
    I1 = I
    RETURN
END
SUBROUTINE CUR3YA( SLP1 , SLPN , SIGMA )
DIMENSION ARCYA(150)
DIMENSION X2P(150) , Y2P(150)
DIMENSION F(150) , S(408)
COMMON/JCHNF/ F
COMMON/JOHNS/ S
COMMON/AYA/ ARCYA
COMMON/YA/ X2F , Y2F , MGF
C  FITS SPLINE==Y2P AS A FUNCTION OF ARCYA
    N= MGP
    NM1 = N-1
    NP1 = N+1
    DELX1 = ARCYA(2)-ARCYA(1)
    DX1 = (Y2P(2)-Y2P(1))/DELX1
    IF (SIGMA.LT.0.) GO TO 5
```

101

```
SLPP1 = SLP1
SLPPN = SLFN
1 SIGMAP = ABS(SIGMA)*(N=1)/(ARCYA(N)=ARCYA(1))
DELS = SIGMAP*DELX1
EXPS = EXP(DELS)
SINHS = .5*(EXPS-1./EXPS)
SINHIN = 1./(DELX1*SINHS)
DIAG1 = SINHIN*(DELS*.5*(EXPS+1./EXPS)-SINHS)
DIAGIN = 1./DIAG1
F(1) = DIAGIN*(DX1-SLPP1)
SPDIAG = SINHIN*(SINHS-DELS)
S(1) = DIAGIN*SPDIAG
IF (N.EQ.2) GO TO 3
DO 2 I = 2,NM1
  DELX2 = ARCYA(I+1)-ARCYA(I)
  DX2= (Y2P(I+1)-Y2P(I))/DELX2
  DELS = SIGMAP*DELX2
  EXPS = EXP(DELS)
  SINHS = .5*(EXPS-1./EXPS)
  SINHIN = 1./(DELX2*SINHS)
  DIAG2 = SINHIN*(DELS*.5*(EXPS+1./EXPS))-SINHS)
  DIAGIN = 1./((DIAG1+DIAG2-SPDIAG*S(I-1)))
  F(I) = DIAGIN*(DX2-DX1-SPDIAG* F(I-1))
  SPDIAG = SINHIN*(SINHS-DELS)
  S(I) = DIAGIN*SPDIAG
  DX1 = DX2
2  DIAG1 = DIAG2
3  DIAGIN = 1./((DIAG1-SPDIAG*S(NM1)))
  F(N) = DIAGIN*(SLPPN=DX2-SPDIAG* F(NM1))
DO 4 I = 2,N
  IBAK = NP1-I
```

```
4      F(IBAK) = F(IEAK)= S(IEAK)* F( IBAK+1)
      RETURN
5  IF (N.EQ.2) GO TO 6
      DELX2 = ARCYA(3)=ARCYA(2)
      DELX12 = ARCYA(3)=ARCYA(1)
      C1 = -(DELX12+DELX1)/DELX12/DELX1
      C2 = DELX12/DELX1/DELX2
      C3 = -DELX1/DELX12/DELX2
      SLPP1 = C1*Y2F(1)+ C2*Y2P(2)+ C3*Y2P(3)
      DELN = ARCYA(N)=ARCYA(NM1)
      DELNM1 = ARCYA(NM1)=ARCYA(N=2)
      DELNN = ARCYA(N)=ARCYA(N=2)
      C1 = (DELNN+DELN)/DELNN/DELN
      C2 = -DELNN/DELN/DELNM1
      C3 = DELN/DELNN/DELNM1
      SLPPN = C3*Y2P(N=2)+ C2*Y2P(NM1)+ C1*Y2P(N)
      GO TO 1
6  F(1) = 0.
      F(2) = 0.
      RETURN
      END
      FUNCTION CURSYA( T , SIGMA , IT )
      DIMENSION ARCYA(150)
      DIMENSION X2P(150) , Y2P(150)
      DIMENSION F(150)
      COMMON/JCHNF/ F
      COMMON/AYA/ ARCYA
      COMMON/YA/ X2P , Y2P , MQP
C THIS FUNCTION INTERPOLATES THE SEAWARD BCRY AT A GIVEN VALUE FOR
C THE ARCLENGTH. THE VALUE RETURNED IN CUREYA IS THE VALUE OF Y
C AT ITS PARTICULAR ARCLENGTH. SUBROUTINE CUR3YA MUST BE CALLED BEFORE.
```

```

N= MQP
S = ARCYA(N)=ARCYA(1)
SIGMAP = ABS(SIGMA)*(N=1)/S
IF (IT.EQ.1) I1 = 2
1 DO 2 I = I1,N
    IF( ARCYA(I)=T ) 2,2,3
2  CONTINUE
I = N
3  IF( ARCYA(I=1) .LE. T .OR. T .LE. ARCYA(1) ) GO TO 4
I1 = 2
GO TO 1
4 DEL1 = T-ARCYA(I=1)
DEL2 = ARCYA(I)=T
DELS = ARCYA(I)-ARCYA(I=1)
EXPS1 = EXP(SIGMAP*DEL1)
SINHD1 = .5*(EXPS1=1./EXPS1)
EXPS = EXP(SIGMAP*DEL2)
SINHD2 = .5*(EXPS=1./EXPS)
EXPS = EXPS1*EXPS
SINHS = .5*(EXPS=1./EXPS)
SINHS = .5*(EXPS=1./EXPS)
CUR5YA= ( F(I)*SINHD1+ F(I=1)*SINHD2)/SINHS+( (Y2P(I)=F(I))*DEL1 +
9 (Y2P(I=1) -F(I=1))*DEL2 )/DELS
I1 = I
RETURN
END
FUNCTION SINH(X)
Y=X
S=EXP(X)-EXP(Y)
SINH= S/2.
RETURN

```

```
END
FUNCTION COSH(X)
Y=-X
S= EXP(X)+EXP(Y)
COSH= S/2.
RETURN
END
```

APPENDIX B  
FORTRAN Listing of Program GRID

```
C PROGRAM GRID. DETERMINES THE COMPUTING GRID DATA FOR INPUT
C TO PROGRAM SSURGE. IT IS ASSUMED THAT THE CONFORMAL MAPPING
C OF THE REGION IS COMPLETE ( PROGRAM CONFORM ) WITH OUTPUT OF THE
C COEFFS.
C
C GRID FOR HURRICANE CARLA
C
DIMENSION COB(200) , COC(200)
DIMENSION X2(1000) , Y2(1000)
DIMENSION X2P(1000) , Y2P(1000)
DIMENSION ATXA(1000) , ATYA(1000) , ATXB(100) , ATYB(1000)
DIMENSION XI(47) , ETA(15) , X(47,15) , Y(47,15) , Z(47,15)
DIMENSION SX(100) , SY(100)
DIMENSION A(5),B(5),C(5),IIC(5)
COMMON /YE/ X2,Y2,MQ
COMMON /YA/ X2P,Y2P,MQP
COMMON/XIETXY/ XI,ETA,X,Y
COMMON/SXY/ SX,SY
COMMON/ED/ XKBETA,BZPBT,BZMBT
COMMON /FORIA/ COB,COC,BZRC,XK,NMAX
C
C READ NUMBER OF COEFFS, NUMBER OF XI LINES+2, NUMBER OF ETA LINES,
C VALUE OF DELTA S STAR IN NM, FIRST VALUE OF S STAR IN NM, VALUE OF
C DELTA T STAR IN MINUTES, NUMBER OF DEPTHS, NUMBER OF POINTS
C
READ(5,1)NMAX,NUMXI,NUMETA,DELSS,SSTRT,DELT,ND,NS
1 FORMAT(3I4, 3F5.1, 2I4)
C
C
SLMAP=60./51.
XLAMDA= 360.
```

```
XIDUM= 360.0*26.0/33.0
PI = 3.141593
XK = PI/XLANDA
SIGMA = =1.
G= 32.2
NUMET1= NUMETA=1
NUMXI1= NUMXI=1
NUMXI2= NUMXI=2
C
      READ (5,27) BETA,BZRO
      DO 26 N=1,NMAX
         READ (5,27) COE(N),COC(N)
26  CONTINUE
27  FORMAT (2E14.7)
      BZPBT= BZRO + BETA
      BZMBT= BZRO - BETA
      XKBETA= XK*BETA
C
C*** DETERMINE SHORELINE COORDINATES (XI,ETA) IN TERMS OF A STRETCHED SHELF
C*** COORDINATE SYSTEM (S*,T*).
C*** S* AXIS IS PARALLEL TO XI AND IS REPRESENTATIVE OF COASTLINE
C*** ARCLENGTH.
C*** T* AXIS IS PARALLEL TO ETA AND IS REPRESENTATIVE OF LONG WAVE
C*** TRAVEL TIME.
C
C
C*** DETERMINE TRANSFORM GENERATED COASTLINE AND SEA BDRY
C
      N3=1
      N4= 121
      DELXI= XLANDA/(N4-1)
```

```
DO 85 I=N3,N4
XID = (I-1)*DELXI
CALL TRAN( XC,YC,XS,YS, XID   )
ATXB(I)= XC
ATYB(I)= YC
ATXA(I)= XS
ATYA(I)= YS
85 CONTINUE
WRITE(6,29)
WRITE(6,39)
WRITE(6,79)
CALL XUT( ATXB , ATYB , N3,N4 )
WRITE(6,29)
WRITE(6,49)
WRITE(6,69)
WRITE(6,79)
CALL XUT( ATXA , ATYA , N3 , N4 )

C
C*** DETERMINE TRANSFORM GENERATED ARCLENGTH OF COASTLINE AND SEA BDRY
C*** AS A FUNCTION OF (EVENLY SPACED) XI.
C
Y2(1)= 0.
X2(1)= 0.
X2P(1)= 0.
DO 91 I=N3,N4
Y2(I)= (I-1)*DELXI
II= I+1
IF( I .EQ. N4 ) GO TO 91
DELX= ABS( ATXB(II)-ATXB(I) )
DELY= ABS( ATYB(II)-ATYB(I) )
X2(II)= SLMAP*SQRT( DELX*DELX + DELY*DELY )
```

```
X2(II)= X2(II) + X2(I)
DELX= ABS( ATXA(II)-ATXA(I) )
DELY= ABS( ATYA(II)-ATYA(I) )
X2P(II)= SLMAP*SQRT( DELX*DELX + DELY*DELY )
X2P(I)= X2P(II) + X2P(I)
91 CONTINUE
WRITE(6,29)
WRITE(6,79)
WRITE(6,89)
CALL XUT( X2,Y2,N3,N4 )
WRITE(6,29)
WRITE(6,79)
WRITE(6,99)
CALL XUT( X2P,Y2,N3,N4 )

C
29 FORMAT( 1H1,/,46X,39HREGION---LAGUN MADRE TC MARSH ISLAND      )
39 FORMAT( //,55X,21HCOASTLINE COORDINATES,//)
40 FORMAT(   62X,8HORIGINAL,////)
49 FORMAT( //,51X,28HSEAWARD BOUNDARY COORDINATES)
50 FORMAT( 53X,24H( 100 FM DEPTH CONTOUR ),//)
59 FORMAT( 54X,23HINTERPOLATION BY SPLINE,////)
69 FORMAT( // )
79 FORMAT( 56X,19HTRANSFORM GENERATED ,/// )
89 FORMAT( 50X,30HARCLENGTH,XI    ALONG COASTLINE,///           )
99 FORMAT( 47X,37HARCLENGTH,XI    ALONG SEAWARD BOUNDARY,///       )

C
MQ= N4
CALL CURV1( SLP1,SLPN,SIGMA )

C
C DETERMINE APPROPRIATE VALUES OF XI SUCH THAT DELTA S* IS CONSTANT .
WRITE(6,29)
```

```
      WRITE(6,113)
113  FORMAT( //,53X,33HSTRETCHED SHELF COORDINATE SYSTEM
         ,9/,39X,52HEVENLY SPACED IN S=STAR (DISTANCE PARALLEL TO COAST) ,
         ,9/,39X,57HAND EVENLY SPACED IN T=STAR (TRAVEL TIME NORMAL TO COAST)
         ,9,/// )
C   TOTAL OF IIC(IRG)=NUM OF INCREMENTS OF DEL S*. REMEMBER THERE ARE
C   TWO EXTRA XI LINES= ONE AT EACH END OF THE GRID.
C   USER SUPPLIES EXPANSION COEFFS A,B,C FOR EACH REGION OF THE CURVE
C   S=PAR= A+B*(S*)**C
C
        IRG= 5
        IIC(1)= 10
        IIC(2)= 4
        IIC(3)= 7
        IIC(4)= 9
        IIC(5)= 16
        A(1)= 0.6058736E+03
        E(1)= -0.5479725E+05
        C(1)= -0.9044300E+00
        A(2)= -0.1395315E+03
        E(2)= 1.5
        C(2)= 1.
        A(3)= 0.4788464E+03
        E(3)= -0.3038714E+07
        C(3)= -0.1707694E+01
        A(4)= 0.
        E(4)= 1.
        C(4)= 1.
        A(5)= 0.2870103E+03
        E(5)= 0.4707750E-11
        C(5)= 0.5160192E+01
```

```
      WRITE(6,114)
114  FORMAT(29X,11HD S PL/D S* ,9X,6HS=STAR,7X,10HS=PARALLEL,11X,2HXI,
9//)
C
      K2= 0
      K1= 1
      DO 120 I=1,IRG
      A1= A(I)
      B1= B(I)
      C1= C(I)
      WRITE(6,115) I,A1,B1,C1
115  FORMAT(1X,7HREGION ,I1,3X,6HA,B,C= ,3(2X,F14.7),//)
      K2= K2+IIC(I)
      K3= K2+1
      DO 119 J=K1,K3
      K=J
      XCUM= SSSTRT + (J-1)*DELSS
      CALL CURV9( XCUM,SDUM,DDUM,A1,B1,C1 )
      XI(J)= CURV2( SDUM,SIGMA,K )
      WRITE(6,117) DIDUM,XDLN,SDUM,XI(J),J
117  FORMAT( 30X,F8.4,10X,F6.1,9X,F6.1,9X,F6.1 , 4X,I2 )
119  CONTINUE
      K1= K3
120  CONTINUE
C
C*** XI VALUES NOW KNOWN FOR EVENLY SPACED INCREMENTS OF S*.
C
C
C*** PROCEEDING TO FIND ETA VALUES FOR EVENLY SPACED INCREMENTS OF T*.
C
C*** DETERMINE ARCLENGTH ALONG A PARTICULAR ISOCLINE OF XI ( XIDUM ).
```

```

C
      DELETA= 2.0*BETA/(NS-1)
      X2(1)= 0.
      WRITE(6,29)
      WRITE(6,79)
      WRITE(6,999) XIDUM
999   FORMAT( 41X,33HARCLENGTH (N NI),ETA     ALCNC X I=      ,F8.3,///      )
      DO 94 J=1,NS
      Y2(J) = -EETA + (J-1)*DELETA
      CALL TRAN1( XDUM,YDUM, XIDUM,Y2(J)  )
      X2P(J)= XDUM
      Y2P(J)= YDUM
      IF( J .EQ. 1 ) GO TO 94
      JJ= J-1
      DELX= ABS( X2P(J)-X2P(JJ) )
      DELY= ABS( Y2P(J)-Y2P(JJ) )
      X2(J)= X2(JJ) + SLMAP*SQRT( DELX*DELX + DELY*DELY )
94    CONTINUE
      DO 98 I=1,NS
      WRITE(6,97) I,X2(I),Y2(I) ,X2P(I),Y2P(I)
97    FORMAT( 42X,I2,2X,F6.1,1H,,F7.3,10X,F6.1,1H,,F6.1
98    CONTINUE
C
      MQ= NS
      CALL CURV1( SLP1,SLPN,SIGMA )
C
C      DETERMINE APPROPRIATE VALUES OF ETA SUCH THAT DELTA T* IS CONSTANT.
C
C      READ WATER DEPTH (FM) AT ND PLACES EVENLY SPACED ALONG XI=XIDUM.
      ND1= ND-1
      DELS= X2(NS)/ND1

```

121

```
Y2P(1)= 0.
DO 123 I=1,ND
READ(S,124) SY(I)
J=I+1
IF( I .EQ. ND ) GO TO 123
Y2P(J)= Y2P(I) + DELS
123 CONTINUE
124 FORMAT( F8.2 )
C DETERMINE LONG WAVE TRAVEL TIME (MIN) ALONG XI=XIDUM.
X2P(1)= 0.0
DO 128 I=1,ND1
J=I+1
F1= 1.0/SQRT( G*SY(I)*6.0 )
F2= 1.0/SQRT( G*SY(J)*6.0 )
X2P(J)= X2P(I) + (F1+F2)*(Y2P(J)-Y2P(I))*50.0
128 CONTINUE
TIM= X2P(ND)
C
MQP= ND
CALL CURV3( SLP1,SLPN,SIGMA )
C
WRITE(6,29)
WRITE(6,131) XIDUM
131 FORMAT( 70X,3HXI=,F6.1,/,13X,8HD T/D T*,12X,6HT=STAR ,6X,4HTIME ,
96X,8HS=NORMAL ,8X,3HETA ,10X, 5HDEPTH, // )
DO 133 I=1,ND
133 WRITE(6,134) X2P(I),Y2P(I),SY(I)
134 FORMAT( 43X,F6.1,6X,F6.1,21X,F6.1 )
C
C TOTAL OF IIC(IRG)=NUM OF INCREMENTS OF DEL=T*.
C EXPANSION CURVE== T= A+E*(T*)**C
```

C  
IRG= 2  
IIC( 1)= 4  
IIC( 2)= 10  
TIM1= IIC(1)\*DELT  
TIM2= IIC(2)\*DELT  
A( 1)= 0.  
C( 1)= TIM1/(TIM-TIM2)  
B( 1)= (TIM-TIM2)\*TIM1\*\*(-C( 1))  
A( 2)= TIM-TIM1-TIM2  
B( 2)= 1.  
C( 2)= 1.  
IF( IIC(1)+1 .EQ. NUMETA ) DELTT= TIM/(NUMETA-1)  
IF( IIC(1)+1 .EQ. NUMETA ) B( 1)=1.  
IF( IIC(1)+1 .EQ. NUMETA ) C( 1)=1.  
WRITE(6,29)  
WRITE(6,113)  
WRITE(6,131) XIDUM  
K2= 0  
K1= 1  
DO 146 I=1,IRG  
A1= A( I )  
B1= B( I )  
C1= C( I )  
WRITE(6,115) I,A1,B1,C1  
K2= K2+IIC(I)  
K3= K2+1  
DO 145 J=K1,K3  
K=J  
WDUM= ( J-1)\*DELT  
CALL CURV9( WDUM,XDUM,DDUM,A1,B1,C1 )

```
YDUM= CURV4( XDUM,SIGMA, K )
ZDUM= CURV2( YDUM,SIGMA,K )
ETA(J)= ZDUM
IF( J .EQ. 1 ) FTA(J)==BETA
IF( J .EG. NUMETA ) ETA(J)= BETA
WRITE(6,143) DOUN,WDUM,XDUM,YDUM,ZDUM, J
143  FORMAT( 11X,F8.4,12X,F6.1,6X,F6.1,6X,F6.1,7X,F7.3,2X,I2 )
145  CCNTINUE
K1= K3
146  CCNTINUE
C
C      *****
C
C*** THE VALUES OF XI AND ETA ARE NOW DETERMINED PERMITTING OUTPUT/
C*** PUNCH OF ALL COMPUTING GRID DATA.
C      X,Y GRID LOCATIONS
C      SCALE FACTOR MU (XI)
C      SCALE FACTOR NU (ETA)
C      SCALE FACTOR S (X,Y/XI,ETA)
C      COS AND SIN THETA = ORIENTATION OF XI AXIS TO X AXIS
C
C
C*** COMPUTE X,Y COORDINATES OF GRID POINT LOCATIONS
C
CALL SHFCCR(      DELSS,NUMXI,      DELTT,NUMETA, SLMAP )
C
C      PUNCH X,Y COORDINATES HERE AS READ IN PROGRAM SSURGE WITH NAME XX,YY.
C      THESE COORDINATES ARE IN ARRAYS X,Y FOR I=2,3==NUMXI-1
C      AND J=1,2==NUMETA
C
C
```

```
C*** COMPUTE SCALE FACTOR MU
C
C      WRITE(6,200)
200  FORMAT( 1H1,//,50X,35H HURRICANE CARLA COMPUTING GRID DATA      )
      WRITE(6,201)
201  FORMAT( //,60X,15H SCALE FACTOR MU ,//  )
      DO 204 I=2,NUMXI
      J=I-1
      SX(J)= ( X(I+1)-X(I-1) )/2.*DELSS
      WRITE(6,202) J,SX(J)
202  FORMAT( 60X,3H MU( ,I2, 3H)= ,E14.7  )
204  CONTINUE
C
C      PUNCH MU HERE AS READ IN PROGRAM SSURGE WITH NAME DSDXI.
C      MU IS IN ARRAY SX FOR I=1,2--NUMXI=2
C
C
C*** COMPUTE SCALE FACTOR NU
C
C      WRITE(6,200)
      WRITE(6,210)
210  FORMAT( //,60X,15H SCALE FACTOR NU ,//  )
      SY(1)=(ETA(2)-ETA(1))/DELT
      SY(NUMETA)=(ETA(NUMETA)-ETA(NUMET1))/DELT
      DO 214 J=1,NUMETA
      IF( J .EQ. 1 .OR. J .EQ. NUMETA ) GO TO 211
      SY(J)=(ETA(J+1)-ETA(J-1))/2.*DELT
211  WRITE(6,212) J , SY(J)
212  FORMAT( 60X,3H NU( ,I2, 3H)= ,E14.7  )
214  CONTINUE
C
```

```
C PUNCH NU HERE AS READ IN PROGRAM SSURGE WITH NAME DTDET.  
C     NU IS IN ARRAY SY FOR J=1,2--NUMETA.  
C  
C*** COMPUTE SCALE FACTOR F AS READ IN PROGRAM SSURGE WITH NAME S.  
C     RECALL F IS SQRT OF AREA IN X,Y PLANE / MU*NU*DELSS*DELT  
C     FIRST COMPUTE AREA OF EACH QUADRANGLE SUB-DIVIDED INTO  
C     IQUAD INTERVALS BETWEEN XI LINES AND JQUAD INTERVALS BETWEEN  
C     ETA LINES  
C  
IQUAD= 4  
JQUAD= 2  
KCP= 1  
CALL TRAN2( KCP, 0., 0., 0., 0., 0. )  
KOP= 2  
WRITE(6,200)  
WRITE(6,300)  
300 FORMAT(//,60X,14HSCALE FACTOR F    // )  
DO 330 I=1,NUMXI  
DELXI= (XI(I+1)-XI(I))/IQUAD  
FRST= XI(I) +DELXI/2.  
DO 310 K=1,IQUAD  
310 X2(K)= FRST + (K-1)*DELXI  
DO 330 J=1,NUMET1  
DELETA= (ETA(J+1)-ETA(J))/JQUAD  
FRST= ETA(J) + DELETA/2.  
SUM= 0.  
DO 320 K=1,IQUAD  
DO 320 L=1,JQUAD  
DUMETA= FRST + (L-1)*DELETA  
CALL TRAN2( KOP,X2(K),DUMETA,0.,DF1,DF2 )
```

```
SUM= SUM + CF1*CF1 + CF2*CF2
320 CONTINUE
      X(I,J)= SUM*DELX I*DELETA
330 CONTINUE
      FAC1= 2.*DELSS*DELT
      FAC2= 2.*FAC1
      N= NUMETA
      DO 370 I=1,NUMXI2
      Z(I,1)=SQRT((X(I,1)+X(I+1,1))/FAC1/SX(I)/SY(1))
      Z(I,N)= SQRT((X(I,N-1)+X(I+1,N-1))/FAC1/SX(I)/SY(N))
      DO 370 J=2,NUMET1
      Z(I,J)= SQRT((X(I,J)+X(I+1,J)+X(I,J-1)+X(I+1,J-1))/FAC2/SX(I)/SY(J
      9))
370 CONTINUE
      DO 390 J=1,NUMETA
      WRITE(6,381) J
381 FORMAT( //,10X,11HROW NUMBER ,I2,// )
      WRITE(6,382) ( I,Z(I,J),I=1,NUMXI2 )
382 FORMAT(   E(14,2X,E14.7,4X))
      390 CONTINUE
C
C PUNCH F HERE AS READ IN PROGRAM SSURGE WITH NAME S. SCALE FACTOR F
C IS IN ARRAY Z FOR I=1,2==NUMXI2 AND J=1,2==NUMETA
C
C
C*** COMPUTE COS AND SIN THETA = ORIENTATION OF XI AXIS TO X AXIS.
C      RECALL, THETA IS A WEIGHTED AVERAGE.
C
      WRITE(6,200)
      WRITE(6,395)
395 FORMAT( //, 60X, 27HVALUES OF COS AND SIN THETA , // )
```

```

CC 400 I=1,NUMXI
XID= XI(I)
DO 400 J=1,NUMETA
ETAD= ETA(J)
CALL TRAN2( KCF,XID,ETAD, ANGLE,CF1,DF2 )
Z(I,J)= ATAN2( DF2,DF1 )
400 CCNTINUE
N= NUMETA
DO 410 I=2,NUMXI1
K= I-1
X(K,1)= COS((Z(I-1,1)+2.*Z(I,2)+Z(I+1,1)+ 4.*Z(I,1))/8.)
Y(K,1)= SIN((Z(I-1,1)+2.*Z(I,2)+Z(I+1,1)+ 4.*Z(I,1))/8.)
X(K,N)= CCS((Z(I-1,N)+2.*Z(I,N-1)+Z(I+1,N)+4.*Z(I,N))/8.)
Y(K,N)= SIN((Z(I-1,N)+2.*Z(I,N-1)+Z(I+1,N)+4.*Z(I,N))/8.)
DO 410 J=2,NUNET1
X(K,J)= CCS((Z(I-1,J)+Z(I,J-1)+Z(I+1,J)+Z(I,J+1)+4.*Z(I,J))/8.)
Y(K,J)= SIN((Z(I-1,J)+Z(I,J-1)+Z(I+1,J)+Z(I,J+1)+4.*Z(I,J))/8.)
410 CCNTINUE
DO 440 J=1,NUNETA
WRITE(6,381) J
WRITE(6,430) ( I,X(I,J),Y(I,J),I=1,NUMXI2)
430 FORMAT( 5(13.2X,F8.5,1X,F8.5,4X))
440 CONTINUE
C
C PUNCH COS AND SIN THETA HERE AS READ IN FRCGRAM SSURGE WITH
C NAMES COSG AND SING. THESE VALUES ARE IN ARRAYS X AND Y FOR
C I=1,2==NUMXI2 AND J=1,2==NUMETA.
C
STOP
END
SUBROUTINE XUT( X,Y,N1,N2 )

```

```
DIMENSION X(1000) , Y(1000)
M= N2-N1+1
M14= M/4
M4= M14*4
IADD= M-M4
DC 10 I=N1,M14
J= I+M14
K= J+M14
L= K+M14
10 WRITE(6,11) I,X(I),Y(I),J,X(J),Y(J),K,X(K),Y(K),L,X(L),Y(L)
11 FORMAT( 4(6X,I4,2X,F6.1,1X,1H,,1X,F6.1,6X) )
IF( IADD .EQ. 0 ) GC TO 25
IF( IADD .EQ. 1 ) NUM1= N2
IF( IADD .EQ. 2 ) NUM1= N2-1
IF( IADD .EQ. 3 ) NUM1= N2-2
DO 20 I=NUM1,N2
20 WRITE(6,21) I,X(I),Y(I)
21 FORMAT( 105X,I4,2X,F6.1,1X,1H,,1X,F6.1,6X )
25 CONTINUE
RETURN
END .
SUBROUTINE SHRCOR(      DELXI,NUMXI,      DELETA,NUMETA,SLMAP )
DIMENSION XI(47) , ETA(15) , X(47,15) , Y(47,15)
DIMENSION SX(100) , SY(100)
COMMON/XIETXY/ XI,ETA,X,Y
COMMON/SXY/ SX,SY
NUMPG= NUMXI/5
IADD= NUMXI-5*NUMPG
NPAGE= 1
DO 40 I=1,NUMXI
DO 40 J=1,NUMETA
```

129

```
CALL TRAN1( XDUM,YDUM,XI(I),ETA(J) )
X(I,J)= XDUM
Y(I,J)= YDUM
40 CONTINUE
55 CONTINUE
          WRITE(6,900) DELXI , DELETA
N1= NPAGE*5 -4
N2= NPAGE*5
IF( N2 .GT. NUMXI ) GO TO 80
IF( NUMPG .EQ. 0 ) GO TO 80
I1= N1
I2= N1+1
I3= N1+2
I4= N1+3
I5= N1+4
WRITE(6,903) XI(I1),XI(I2),XI(I3),XI(I4),XI(I5),I1,I2,I3,I4,I5
DO 70 K=1,NUMETA
J= NUMETA-K+1
JJ= J-1
DO 60 I=N1,N2
IF( I .EQ. NUMXI ) GO TO 62
II= I+1
DELX= ABS( X(II,J)-X(I,J) )
DELY= ABS( Y(II,J)-Y(I,J) )
SX(I)= SLMAP*SQRT( DELX*DELX + DELY*DELY )
60 CONTINUE
62 CONTINUE
IF( IADD .EQ. 0 .AND. N2 .EQ. NUMXI ) GO TO 64
WRITE(6,904) ETA(J),X(I1,J),Y(I1,J),SX(I1),X(I2,J),Y(I2,J),SX(I2),
9X(I3,J),Y(I3,J),SX(I3),X(I4,J),Y(I4,J),SX(I4),X(I5,J),Y(I5,J) ,
9SX(I5), J
```

```
      GC TO 66
64  CONTINUE
      WRITE(6,905) ETA(J),X(I1,J),Y(I1,J),SX(I1),X(I2,J),Y(I2,J),SX(I2),
      9X(I3,J),Y(I3,J),SX(I3),X(I4,J),Y(I4,J),SX(I4),X(I5,J),Y(I5,J), J
66  CONTINUE
      IF( K .EQ. NUMETA ) GO TO 70
      DO 68 I=N1,N2
      DELX= ABS( X(I,JJ)-X(I,J) )
      DELY= ABS( Y(I,JJ)-Y(I,J) )
      SY(I)= SLNAP*SQRT( DELX*DELX + DELY*DELY )
68  CONTINUE
      WRITE(6,906) SY(I1),SY(I2),SY(I3),SY(I4),SY(I5)
70  CONTINUE
      IF( IADD .EQ. 0 .AND. N2 .EQ. NUMXI ) GO TO 150
      NPAGE= NPAGE + 1
      GC TO 55
80  CONTINUE
      IF( IADD .EQ. 1 ) GO TO 100
      IF( IADD .EQ. 2 ) GO TO 110
      IF( IADD .EQ. 3 ) GC TO 120
      IF( IADD .EQ. 4 ) GC TO 130
100 CCNT INUE
      I1= N1
      I= I1
      WRITE(6,910) XI(I1),I1
      DO 102 K=1,NUMETA
      J= NUMETA-K+1
      JJ= J-1
      WRITE(6,911) X(I1,J),Y(I1,J)
      IF( K .EQ. NUMETA ) GO TO 102
      DELX= AES( X(I,JJ)-X(I,J) )
```

|3|

```
      DELY= ABS( Y(I,JJ)-Y(I,J) )
      SY(I)= SLMAP*SQRT( DELX*DELX + DELY*DELY )
      WRITE(6,912) SY(I1)
102   CCNTINUE
      GO TO 150
110   CONTINUE
      I1= N1
      I2= N1+1
      WRITE(6,914) XI(I1),XI(I2),I1,I2
      DO 114 K=1,NUMETA
      J= NUMETA-K+1
      JJ= J+1
      DELX= ABS( X(I2,J)-X(I1,J) )
      DELY= ABS( Y(I2,J)-Y(I1,J) )
      SX(I1)= SLMAP*SQRT( DELX*DELX + DELY*DELY )
      WRITE(6,915) X(I1,J),Y(I1,J),SX(I1),X(I2,J),Y(I2,J)
      IF( K .EQ. NUMETA ) GO TO 114
      DC 112 I=I1,I2
      DELX= ABS( X(I,JJ)-X(I,J) )
      DELY= ABS( Y(I,JJ)-Y(I,J) )
      SY(I)= SLMAP*SQRT( DELX*DELX + DELY*DELY )
112   CCNTINUE
      WRITE(6,916) SY(I1),SY(I2)
114   CCNTINUE
      GO TO 150
120   CCNTINUE
      I1= N1
      I2= N1+1
      I3= N1+2
      WRITE(6,917) XI(I1),XI(I2),XI(I3),I1,I2,I3
      DO 128 K=1,NUMETA
```

132

```
J= NUMETA-K+1
JJ= J=1
DO 122 I=I1,I3
IF( I .EQ. NUMXI ) GO TO 124
II= I+1
DELX= ABS( X(II,J)-X(I,J) )
DELY= ABS( Y(II,J)-Y(I,J) )
SX(I)= SLMAP*SQRT( DELX*DELX + DELY*DELY )
122 CONTINUE
124 CONTINUE
WRITE(6,918) ETA(J),X(I1,J),Y(I1,J),SX(I1),X(I2,J),Y(I2,J),SX(I2),
9X(I3,J),Y(I3,J), J
IF( K .EQ. NUMETA ) GO TO 128
DO 126 I=I1,I3
DELX= ABS( X(I,JJ)-X(I,J) )
DELY= ABS( Y(I,JJ)-Y(I,J) )
SY(I)= SLMAP*SQRT( DELX*DELX + DELY*DELY )
126 CCNTINUE
WRITE(6,919) SY(I1),SY(I2),SY(I3)
128 CONTINUE
GO TO 150
130 CONTINUE
I1= N1
I2= N1+1
I3= N1+2
I4= N1+3
WRITE(6,921) XI(I1),XI(I2),XI(I3),XI(I4),I1,I2,I3,I4
DO 138 K=1,NUMETA
J= NUMETA-K+1
JJ= J=1
DO 132 I=I1,I4
```

```
IF( I .EQ. NUNXI ) GO TO 134
II= I+1
DELX= ABS( X(II,J)-X(I,J) )
DELY= ABS( Y(II,J)-Y(I,J) )
SX(I)= SLMAP*SQRT( DELX*DELX + DELY*DELY )
132 CONTINUE
134 CONTINUE
      WRITE(6,922) ETA(J),X(II,J),Y(II,J),SX(II),X(I2,J),Y(I2,J),SX(I2),
      9X(I3,J),Y(I3,J),SX(I3),X(I4,J),Y(I4,J), J
      IF( K .EQ. NUMETA ) GO TO 138
      DO 136 I=II,I4
      DELX= ABS( X(I,JJ)-X(I,J) )
      DELY= ABS( Y(I,JJ)-Y(I,J) )
      SY(I)= SLMAP*SQRT( DELX*DELX + DELY*DELY )
136 CONTINUE
      WRITE(6,923) SY(II),SY(I2),SY(I3),SY(I4)
138 CONTINUE
150 CONTINUE
900 FFORMAT( 1H1,4EX,39FRECTION---LAGUNA MADRE TO MARSH ISLAND ,
      9// ,55X,21HSHCRELINE COORDINATES ,
      9//,58X,16HFURRICANE CARLA
      9// ,46X,10HDELTAT=S*= ,F6.3,6X,10HDELTAT=T*= ,F6.3, // )
903 FORMAT( 2X,5(14X,3HXI=,F7.2),/,5(19X,1H(,I2,1H),1X),// )
904 FORMAT( 1X,4HETA=,F7.3,2X,F6.1,1H,,F6.1
      94(3X,F4.1,4X,F6.1,1H,,F6.1), 2X,F4.1,/ ,5X,1H(,I2,1H)
      905 FORMAT( 1X,4HETA=,F7.3,2X,F6.1,1H,,F6.1
      94(3X,F4.1,4X,F6.1,1H,,F6.1), / ,5X,1H(,I2,1H)
      906 FORMAT( 5(18X,F4.1,2X),/
      910 FORMAT( 16X,3HXI=,F7.2,/ ,19X,1H(,I2,1H),// )
      911 FFORMAT( 14X,F6.1,1H,,F6.1,/
      912 FORMAT( 18X,F4.1,/  
)
```

```

914 FORMAT( 2X,2(14X,3HXI=F7.2),/ ,2(19X,1H(,I2,1H),1X),// )
915 FORMAT( 14X,F6.1,1H,,F6.1,3X,F4.1,4X,F6.1,1H,,F6.1,/ )
916 FORMAT( 2(18X,F4.1,2X),/ )
917 FORMAT( 2X,3(14X,3HXI=F7.2),/ ,3(19X,1H(,I2,1H),1X),// )
918 FORMAT( 1X,4HETA=,F7.3,2X,F6.1,1H,,F6.1
92  (3X,F4.1,4X,F6.1,1H,,F6.1), / ,5X,1H(,I2,1H) )
919 FORMAT( 3(18X,F4.1,2X),/ )
921 FORMAT( 2X,4(14X,3HXI=F7.2),/ ,4(19X,1H(,I2,1H),1X),// )
922 FORMAT( 1X,4HETA=,F7.3,2X,F6.1,1H,,F6.1
93  (3X,F4.1,4X,F6.1,1H,,F6.1), / ,5X,1H(,I2,1H) )
923 FORMAT( 4(18X,F4.1,2X),/
      RETURN
      END
      SUBROUTINE TRAN( XCST,YCST,XSEA,YSEA,XI )
      DIMENSION COB(200), CCC(200)
      COMMON/ED/ XKBETA,BZPBT,BZMBT
      COMMON/FCRIA/ CCB,CCC,BZRC,XK,NMAX
      XCST= 0.
      YCST= 0.
      XSEA= 0.
      YSEA= 0.
      XKXI= XK*XI
      DO 10 N=1,NMAX
      XN=N
      XNKBT= XN*XKEET
      XNKXI= XN*XKXI
      CSH= COSH( XNKBT )
      SNH= SINH( XNKBT )
      A= COC(N)*CSH
      B= COB(N)*SNH
      C= CCC(N)*SNH

```

```
C= CCB(N)*CSH
E= SIN( XNKXI )
F= COS( XNKXI )
XCST= XCST + (A+E)*E
XSEA= XSEA + (A=B)*E
YCST= YCST + ( D+C)*F
YSEA= YSEA + ( D=C)*F
10 CONTINUE
XCST= XCST + XI
XSEA= XSEA + XI
YCST= YCST + BZPBT
YSEA= YSEA + EZMBT
RETURN
END
SUBROUTINE TRAN1( XDUM , YDUM , XI , ETA )
DIMENSION CCB(200), CCC(200)
COMMON/FORIA/ CCB,CCC,BZRC,XK,NMAX
XDUM= 0.
YDUM= 0.
XXXI= XK*XI
XKEta= XK*ETA
DO 10 N=1,NMAX
XN=N
XNKBT= XN*XKEta
XNKXI= XN*XXXI
CSH= CSH( XNKBT )
SNH= SINH( XNKBT )
A= COC(N)*CSH
B= COB(N)*SNH
C= COC(N)*SNH
D= COB(N)*CSH
```

```
E= SIN( XNKXI )
F= COS( XNKXI )
XDUM= XDUM + (A+B)*E
YDUM= YDUM + ( C+C)*F
10  CONTINUE
XDUM= XDUM + XI
YDUM= YDUM + EZRO + ETA
RETURN
END
SUBROUTINE TRAN2( KOP,XI,ETA,ANGLE,DXDXI,DYDXI )
DIMENSION COB(200) , CCC(200) , XNK(200)
COMMON/FORIA/CCB,COC,BZRC,XK,NMAX
IF(KOP.EQ.1) GO TO 100
GO TO 200
100 DO 105 N=1,NMAX
XN= N
105 XNK(N)= XN*XK
GO TO 999
200 DXDXI= 1.
DYDXI= 0.
DO 205 N=1,NMAX
XNKETA= XNK(N)*ETA
XNKXI= XNK(N)*XI
SNH= SINH( XNKETA )
CSH= CCSH( XNKETA )
DXDXI= DXDXI+XNK(N)*(COB(N)*SNH+CCC(N)*CSH)*CCS(XNKXI)
DYDXI= DYDXI=XNK(N)*(COB(N)*CSH+CCC(N)*SNH)*SIN(XNKXI)
205 CCNTINUE
IF( KOP.EQ.2) GO TO 999
ANGLE= ATAN2(DYDXI,DXDXI)
999 RETURN
```

```
END
SUBROUTINE CURV9( XCUM,Y,Y1,A,B,C )
Z= ABS( XCUM )
W= ABS( C=1. )
Y1= B
Y= A+B*(XCUM)**C
IF( Z .LT. 0.0001 .OR. W .LT. 0.001 ) GO TO 10
Y1= B*C*(XCUM)**(C=1.)
10 CONTINUE
RETURN
END
SUBROUTINE CURV1( SLP1 , SLPN , SIGMA )
C
DIMENSION X2(1000) , Y2(1000)
DIMENSION R(800) , S(800)
COMMON/YB/ X2, Y2 , MG
COMMON/JOHN1/ R
COMMON/JOHN2/ S
C
C*****THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO
C COMPUTE AN INTERPOLATORY SPLINE UNDER TENSION THROUGH
C A SEQUENCE OF FUNCTIONAL VALUES. THE SLOPES AT THE TWO
C ENDS OF THE CURVE MAY BE SPECIFIED OR OMITTED. FOR ACTUAL
C COMPUTATION OF POINTS ON THE CURVE IT IS NECESSARY TO CALL
C THE FUNCTION CURV2.
C
C*****ON INPUT--
```

```
C
C   N= MG IS THE NUMBER OF VALUES TO BE INTERPOLATED ( .GE. 2 )
C   X2 IS AN ARRAY OF N INCREASING ABCISSAE OF THE FUNCTIONAL VALUES
C   Y2 IS AN ARRAY OF N ORDINATES OF THE VALUES
C     I.E. Y2(K) IS THE FUNCTIONAL VALUES CORRESPONDING TO X2(K).
C   R IS AN ARRAY OF LENGTH AT LEAST N ( PERMANENT STORAGE )
C   S IS AN ARRAY OF LENGTH AT LEAST N ( SCRATCH STORAGE ).
```

C

```
C   SLP1 AND SLPN CONTAIN THE DESIRED VALUES FOR THE FIRST
C   DERIVATIVE OF THE CURVE AT X2(1) AND X2(N), RESPECTIVELY.
C   IF THE QUANTITY SIGMA IS NEGATIVE THESE VALUES WILL BE
C   DETERMINED INTERNALLY AND THE USER NEED ONLY FURNISH
C   PLACE-HOLDING PARAMETERS FOR SLP1 AND SLPN. SUCH PLACE-
C   HOLDING PARAMETERS WILL BE IGNORED BUT NOT DESTROYED.
```

C

```
C   SIGMA CONTAINS THE TENSION FACTOR. THIS IS NON-ZERO AND
C   INDICATES THE CURVINESS DESIRED. IF ABS(SIGMA) IS NEARLY
C   ZERO (E.G. .001) THE RESULTING CURVE IS APPROXIMATELY A
C   CUBIC SPLINE. IF ABS(SIGMA) IS LARGE (E.G. 50.) THE
C   RESULTING CURVE IS NEARLY A POLYGONAL LINE. THE SIGN
C   OF SIGMA INDICATES WHETHER THE DERIVATIVE INFORMATION
C   HAS BEEN INPUT OR NOT. IF SIGMA IS NEGATIVE THE ENDPOINT
C   DERIVATIVES WILL BE DETERMINED INTERNALLY. A STANDARD
C   VALUE FOR SIGMA IS APPROXIMATELY 1. IN ABSOLUTE VALUE.
```

C

C

```
C   CN OUTPUT==
```

C

```
C   R CONTAINS VALUES PROPORTIONAL TO THE SECOND DERIVATIVE
C   OF THE CURVE AT THE GIVEN NODES ( X2,Y2 ) .
```

C

```
C  
C      N (NG), X2, Y2, SLF1, SLPN, SIGMA ARE UNALTERED.  
C  
C  
C  
      N = MG  
      NM1 = N-1  
      NP1 = N+1  
      DELX1 = X2(2)-X2(1)  
      DX1 = (Y2(2)-Y2(1))/DELX1  
C  
C      DETERMINE SLCPFS IF NECESSARY  
C  
      IF (SIGMA.LT.0.) GO TO 5  
      SLPP1 = SLF1  
      SLPPN = SLPN  
C  
C      DENORMALIZE TENSION FACTOR  
C  
      1  SIGMAP = AES(SIGMA)*(N-1)/(X2(N)-X2(1))  
C  
C      SET UP RIGHT HAND SIDE AND TRIDIAGONAL SYSTEM FOR R AND  
C      PERFORM FORWARD ELIMINATION  
C  
      DELS = SIGMAP*DELX1  
      EXPS = EXP(DELS)  
      SINHS = .5*(EXPS-1./EXPS)  
      SINHIN = 1./(DEFLX1*SINHS)  
      DIAG1 = SINHIN*(DELS*.5*(EXPS+1./EXPS)-SINHS)  
      DIAGIN = 1./DIAG1  
      R(1) = DIAGIN*(DX1-SLPP1)
```

```
SPDIAG = SINHIN*(SINHS-DELS)
      S(1) = DIACIN*SPDIAG
C
      IF (N.EQ.2) GO TO 3
C
      DO 2 I = 2,NM1
        DELX2 = X2(I+1)-X2(I)
        DX2= (Y2(I+1)-Y2(I))/DELX2
        DELS = SIGMAP*DELX2
        EXP5 = EXP(DELS)
        SINHS = .5*(EXP5-1./EXP5)
        SINHIN = 1./(DELX2*SINHS)
        DIAG2 = SINHIN*(DELS*(.5*(EXP5+1./EXP5))-SINHS)
        DIAGIN = 1./(DIAG1+DIAG2-SPDIAG*S(I-1))
        R(I) = DIAGIN*(DX2-DX1-SPCIAG*R(I-1))
        SPDIAG = SINHIN*(SINHS-DELS)
        S(I) = DIACIN*SPDIAG
        DX1 = DX2
2     DIAG1 = DIAG2
C
      3 DIAGIN = 1./(DIAG1-SPCIAG*S(NM1))
        R(N) = DIAGIN*(SLPPN-DX2-SPDIAG*R(NM1))
C
C PERFORM BACK SUBSTITUTION
C
      CC 4 I = 2,N
      IBAK = NF1-I
      4     R(IBAK) = R(IBAK)- S(IBAK)* R(IBAK+1)
C
      RETURN
C
```

C  
C 5 IF (N.EQ.2) GO TO 6  
C  
C IF NO DERIVATIVES ARE GIVEN USE SECOND ORDER POLYNOMIAL  
C INTERPOLATION ON INPUT DATA FOR VALUES AT ENDPOINTS.  
C  
DELX2 = X2(3)-X2(2)  
DELX12 = X2(3)-X2(1)  
C1 = -(DELX12+DELX1)/DELX12/DELX1  
C2 = DELX12/DELX1/DELX2  
C3 = -DELX1/DELX12/DELX2  
SLPP1 = C1\*Y2(1) + C2\*Y2(2) + C3\*Y2(3)  
DELN = X2(N)-X2(NM1)  
DELNM1 = X2(NM1)-X2(N-2)  
DELNN = X2(N)-X2(N-2)  
C1 = (DELNN+DELN)/DELNN/DELN  
C2 = -DELNN/DELN/DELNM1  
C3 = DELN/DELNN/DELNM1  
SLPPN = C3\*Y2(N-2) + C2\*Y2(NM1) + C1\*Y2(N)  
GO TO 1  
C  
C IF ONLY TWO POINTS AND NO DERIVATIVES ARE GIVEN, USE  
C STRAIGHT LINE FOR CURVE  
C  
6 R(1) = 0.  
R(2) = 0.  
C  
RETURN  
END  
FUNCTION CURV2( T , SIGMA , IT )  
C

```
CIMENSION X2(1000) , Y2(1000)
DIMENSION R(800)
COMMON/YB/ X2, Y2 , MG
COMMON/JOHN1/ R
C
C***** ****
C THIS FUNCTION INTERPOLATES A CURVE AT A GIVEN POINT
C USING A SPLINE UNDER TENSION. THE SUBROUTINE CURV1 SHOULD
C BE CALLED EARLIER TO DETERMINE CERTAIN NECESSARY
C PARAMETERS.
C
C***** ****
C
C ON INPUT--
C
C      T (X=COORD) CONTAINS A REAL VALUE TO BE MAFFED CNTO THE CURVE
C      N= (MQ) IS THE NUMBER OF POINTS WHICH WERE INTERPOLATED TO
C          DETERMINE THE CURVE.
C      X2,Y2 ARE ARRAYS CCNTAINING THE AECISSAS AND ORCINATES OF THE
C          ORIGINAL INTERPOLATED POINTS.
C      R IS AN ARRAY WITH VALUES FRCPCRTIONAL TO THE SECOND
C          DERIVATIVE OF THE CURVE AT THE NCODES (X2,Y2) .
C      SIGMA CONTAINS THE TENSION FACTOR (ITS SIGN IS IGNORED)
C
C      IT IS AN INTEGER SWITCH. IF IT IS NOT 1 THIS INDICATES
C          THAT THE FUNCTION HAS BEEN CALLED PREVICUSLY (WITH N,X2,
C          Y2,R, AND SIGMA UNALTERED) AND THAT THIS VALUE OF T
C          EXCEEDS THE PREVICUS VALUE. WITH SUCH INFORMATION THE
C          FUNCTION IS ABLE TO PERFORM THE INTERPOLATION MUCH MORE
C          RAPIDLY. IF A USER SEEKS TO INTERPOLATE AT A SEQUFNCE
```

```
C OF POINTS, EFFICIENCY IS GAINED BY ORDERING THE VALUES
C INCREASING AND SETTING IT TO THE INDEX OF THE CALL.
C IF IT IS 1 THE SEARCH FOR THE INTERVAL (X2(K),X2(K+1))
C CONTAINING T STARTS WITH K=1.
C
C THE PARAMETERS N (MQ), X2, Y2, R, AND SIGMA SHOULD BE INPUTTED
C UNALTERED FROM THE OUTPUT OF CURV1 AND THESE ARE UNALTERED IN CURV2 .
C
C
C ON OUTPUT ==
C
C CURV2 (=Y=CCORD) CONTAINS THE INTERPOLATED VALUE.
C      FOR T .LT. X2(1), CURV2= Y2(1). FOR T .GT. X2(N),CURV2= Y2(N).
C
C
C      N= MQ
C
C DENORMALIZE SIGMA
C      S = X2(N)-X2(1)
C      SIGMAP = AES(SIGMA)*(N-1)/S
C
C IF IT.NF.1 START SEARCH WHERE PREVIOUSLY TERMINATED,
C OTHERWISE START FROM BEGINNING
C
C      IF ( IT.EQ.1) I1 = 2
C
C SEARCH FOR INTERVAL
C
C      1 DO 2 I = I1,N
C          IF( X2(I)=T ) 2,2,3
C      2  CONTINUE
```

```
C
      I = N
C CHECK TO INSURE CORRECT INTERVAL
C
      3 IF( X2(I-1) .LE. T .OR. T .LE. X2(1) ) GO TO 4
C
C RESTART SEARCH AND RESET II
C ( INPUT "'IT'" WAS INCORRECT )
C
      II = 2
      GO TO 1
C
C SET UP AND PERFORM INTERPOLATION
C
      4 DEL1 = T-X2(I-1)
      DEL2 = X2(I)-T
      DELS = X2(I)-X2(I-1)
      EXPS1 = EXP(SIGMAP*DEL1)
      SINHD1 = .5*(EXPS1-1./EXPS1)
      EXPS = EXP(SIGMAP*DEL2)
      SINHD2 = .5*(EXPS-1./EXPS)
      EXPS = EXPS1*EXPS
      SINHS = .5*(EXPS-1./EXPS)
      CURV2 = ( R(I)*SINHD1+ R(I-1)*SINHD2)/SINHS+((Y2(I)-R(I))*DEL1 +
      9*(Y2(I-1) - R(I-1))*DEL2 )/DELS
C
      II = I
      RETURN
      END
      SUBROUTINE CURVE( SLP1 , SLPN , SIGMA )
      DIMENSION X2P(1000) , Y2P(1000)
```

```
DIMENSION V(800), S(800)
COMMON/JHN2/ S
COMMON/JHN4/ V
COMMON /YA/ X2P,Y2P,MQP
C ****
C THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO
C COMPUTE AN INTERPOLATORY SPLINE UNDER TENSION THROUGH
C A SEQUENCE OF FUNCTIONAL VALUES. THE SLOPES AT THE TWO
C ENDS OF THE CURVE MAY BE SPECIFIED OR OMITTED. FOR ACTUAL
C COMPUTATION OF POINTS ON THE CURVE IT IS NECESSARY TO CALL
C THE FUNCTION CURV4.
C ****
C ON INPUT--
C
C      N= MQP IS THE NUMBER OF VALUES TO BE INTERPOLATED ( .GE. 2 )
C      X2P IS AN ARRAY OF N INCREASING ABSISSAE OF THE FUNCTIONAL VALUES
C      Y2P IS AN ARRAY OF N ORDINATES OF THE VALUES
C      V IS AN ARRAY OF LENGTH AT LEAST N (PERMANENT STORAGE )
C      S IS AN ARRAY OF LENGTH AT LEAST N ( SCRATCH STORAGE )..
C
C ON OUTPUT--
C
C      V CONTAINS VALUES FURTHER NORMAL TO THE SECOND DERIVATIVE
C      OF THE CURVE AT THE GIVEN NODES ( X2P,Y2P )
C
C      N(MQP), X2P,Y2F,SLP1, SLPN, SIGMA ARE UNALTERED.
C
```

```
N= MCP
NM1 = N=1
NPI = N+1
DELX1 = X2P(2)-X2P(1)
DX1 = (Y2F(2)-Y2P(1))/DELX1
IF (SIGMA.LT.0.) GO TO 5
SLPP1 = SLP1
SLPFN = SLFN
1 SIGMAP = ABS(SIGMA)*(N=1)/(X2P(N)-X2P(1))
DELS = SIGMAP*DELX1
EXPS = EXP(DELS)
SINHS = .5*(EXPS-1./EXPS)
SINHIN = 1./(DELX1*SINHS)
DIAG1 = SINHIN*(DELS*.5*(EXPS+1./EXPS)-SINHS)
DIAGIN = 1./DIAG1
V(1) = DIAGIN*(DX1-SLPP1)
SPDIAG = SINHIN*(SINHS-DELS)
S(1) = DIAGIN*SPDIAG
IF (N.EQ.2) GO TO 3
DO 2 I = 2,NM1
DELX2 = X2P(I+1)-X2P(I)
DX2= (Y2F(I+1)-Y2P(I))/DELX2
DELS = SIGMAP*DELX2
EXPS = EXP(DELS)
SINHS = .5*(EXPS-1./EXPS)
SINHIN = 1./(DELX2*SINHS)
DIAG2 = SINHIN*(DELS*(.5*(EXPS+1./EXPS))-SINHS)
DIAGIN = 1./((DIAG1+DIAG2-SPDIAG*S(1)))
V(I) = DIAGIN*(DX2-DX1-SPDIAG* V(I-1))
SPDIAG = SINHIN*(SINHS-DELS)
S(I) = DIAGIN*SPDIAG
```

```

DX1 = DX2
2  DIAG1 = SIGMA
3  DIAGIN = 1. / (DIAG1 - SPECIAG * S(NM1))
      V(N) = DIAGIN * (SLPPN = DX2 = SPDIAG * V(NM1))
      DC 4 I = 2, N
      IBAK = NP1 = I
4  V(IBAK) = V(IBAK) - S(IBAK) * V(IBAK + 1)
      RETURN
5  IF (N.EQ.2) GO TO 6
      DELX2 = X2P(3) - X2P(2)
      DELX12 = X2P(3) - X2P(1)
      C1 = -(DELX12 + DELX1) / DELX12 / DELX1
      C2 = DELX12 / DELX1 / DELX2
      C3 = -DELX1 / DELX12 / DELX2
      SLPP1 = C1 * Y2P(1) + C2 * Y2P(2) + C3 * Y2P(3)
      DELN = X2P(N) - X2P(N-2)
      DELNM1 = X2P(NM1) - X2P(N-2)
      DELNN = X2P(N) - X2P(N-2)
      C1 = (DELNN + DELN) / DELNN / DELN
      C2 = -DELNN / DELN / DELNM1
      C3 = DELN / DELNN / DELNM1
      SLPPN = C3 * Y2P(N-2) + C2 * Y2P(NM1) + C1 * Y2P(N)
      GO TO 1
6  V(1) = 0.
      V(2) = 0.
      RETURN
      END
      FUNCTION CURV4( T , SIGMA , IT )
C
      DIMENSION X2P(1000) , Y2P(1000)
      DIMENSION V(800)

```

```
COMMON/JCHN4/ V
COMMON /YA/ X2P,Y2P,MGP
C ****
C THIS FUNCTION INTERPOLATES A CURVE AT A GIVEN POINT
C USING A SPLINE UNDER TENSION. THE SUBROUTINE CURV3 SHOULD
C BE CALLED EARLIER TO DETERMINE CERTAIN NECESSARY PARAMETERS.
C ****
C ON INPUT--
C
C   T (X=CCORD) CONTAINS A REAL VALUE TO BE MAPPED ONTO THE CURVE
C   N= MGP IS THE NUMBER OF POINTS WHICH WERE INTERPOLATED TO
C       DETERMINE THE CURVE.
C   X2P,Y2P ARE ARRAYS CONTAINING THE ABSISSAS AND ORDINATES OF THE
C       ORIGINAL INTERPOLATED POINTS.
C   V IS AN ARRAY WITH VALUES PROPORTIONAL TO THE SECOND
C       DERIVATIVE OF THE CURVE AT THE NODES ( X2P,Y2P )..
C
C   THE PARAMETERS N(MGP),X2P,Y2P, V AND SIGMA SHOULD BE INPUTTED
C   UNALTERED FROM THE OUTPUT OF CURV3 AND THESE ARE UNALTERED IN CURV4 .
C
C ON OUTPUT--
C
C   CURV4 (=Y=CCORD) CONTAINS THE INTERPOLATED VALUE.
C
C   N= MGP
C   S = X2P(N)-X2P(1)
C   SIGMAP = AES(SIGMA)*(N-1)/S
```

```
      IF ( IT.EQ.1) I1 = 2
1 DO 2 I = I1,N
     IF( X2P(I)=T ) 2,2,3
2   CONTINUE
     I = N
3   IF( X2P(I=1) .LE. T .OR. T .LE. X2P(1) ) GC TO 4
     I1 = 2
     GC TO 1
4   DEL1 = T-X2P(I=1)
     DEL2 = X2P(I) - T
     DELS = X2P(I)-X2P(I=1)
     EXPS1 = EXP(SIGMAP*DEL1)
     SINHD1 = .5*(EXPS1-1./EXPS1)
     EXPS = EXP(SIGMAP*DEL2)
     SINHD2 = .5*(EXPS-1./EXPS)
     EXPS = EXPS1*EXPS
     SINHS = .5*(EXPS-1./EXPS)
     CURV4 = ( V(I)*SINHD1+ V(I=1)*SINHD2)/SINHS+((Y2P(I)-V(I))*DEL1 +
9(Y2P(I=1) -V(I=1))*DEL2 )/DELS
     I1 = I
     RETURN
END
FUNCTION COSH(XDAM)
YDAM= -XDAM
COSH= EXP(XDAM) + EXP(YDAM)
COSH= COSH/2.0
RETURN
END
FUNCTION SINH(XDAM)
YDAM= -XDAM
SINH= EXP(XDAM) - EXP(YDAM)
SINH= SINH/2.0
RETURN
END
```

APPENDIX C  
FORTRAN Listing of Program SSURGE

```

C SSURGE
COMMON /BLK2/ C2, CC, C1X, C1ETA, DX, DETA, DELT, F, IM, JM, IMM, JMM, N
COMMON /BLK20/ CORIO, CS, C3, C4, ROT, IMN2, INN3, JMM2
COMMON /BLK3/ GRAV, NMAX, IH1, JH1, IT1, JT1, IH2,
1 JH2, IT2, JT2, IH3, JH3, IT3, JT3, IH4, JH4, IT4, JT4, INC
COMMON /BLK6/ IH5, JH5, IT5, JT5, IH6, JH6, IT6, JT6
COMMON /HURR1/ YRANGE, THIT, XHIT, YHIT, .
3 C10, C11, PHI, .
4 NT1, NT2, .
5 PINF, IX, IY
DATA IM, JM, NMAX, INC/
1 45, 15, 1321, 10/
DATA DX, DETA, DELT, GRAV, F/
1 13081.3, 17441.7, 180.0, 9.8, .0025/
DATA IH1, JH1, IT1, JT1, IH2, JH2, IT2, JT2/
1 18, 15, 18, 14, 10, 15, 10, 14/
DATA IH3, JH3, IT3, JT3, IH4, JH4, IT4, JT4/
1 16, 15, 16, 14, 32, 15, 32, 14/
DATA IH5, JH5, IT5, JT5, IH6, JH6, IT6, JT6/
1 36, 15, 36, 14, 38, 15, 38, 14/

```

```

C
IMM=IM=1
IMM2 = IM=2
IMM3 = IM=3
JMM=JM=1
JMM2=JM=2
CORIO= 6.70875E-05
CC=SQRT(GRAV*DELT)
CS = SQRT(GRAV*3048.)
C2=2.0*DELT
C3 = CORIO*DELT

```

```
C4 = C3*C3
ROT = C3
C1XI=GRAV*DELT/DXI
C1ETA=GRAV*DELT/DETA
C
IX=IM
IY=JM
XHIT=160.
YHIT=168.
THIT=49.35
YRANGE=17.
PHI=20.
PINF=1016.
DEGRAD=3.141593/180.
C10=COS(PHI*DEGRAD)
C11=SIN(PHI*DEGRAD)
C
CALL ZERO
CALL FIELD
DO 100 N=1 ,NMAX
IF((N/2)*2.EQ.N)GO TO 30
CALL WINDF
CALL ELEV
CALL DRAW1
GO TO 100
30 CONTINUE
CALL FLUX
100 CONTINUE
CALL HUV
STOP
END
```

```

SUBROUTINE ZERO
COMMON/BLK1/GRID1(45,15),GRID2(45,15),          S(45,15),HYD1(30
10),HYD2(300),HYD3(300),HYD4(300),UCM1(300),UCM2(300),UCM3(300),UCM
24(300),VCM1(300),VCM2(300),VCM3(300),VCM4(300)
COMMON/BLK2/C2,CC,C1X,I,C1ETA,DXI,DETA,DELT,F,IM,JM,    IMM,JMM, N
COMMON/BLK3/           GRAV,          NMAX, IH1,JH1,IT1,JT1,IH2,
1JH2, IT2,JT2,IH3,JH3,IT3,JT3,IH4,JH4,IT4,JT4,INC
COMMON/BLK5/DSDXI(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6
1(300),VCM5(300),VCM6(300)
COMMON/BLK7/HOBSS1(67),HOBSS2(67),HOBSS3(67),HOBSS4(67),HOBSS5(67),HCBS
16(67)
COMMON/STORM/NO,XEYE,YEYE,P(45,15),WIND(46,15)
COMMON /TRANS/  XX(45,15) ,YY(45,15) ,COSG(45,15),SING(45,15)
COMMON /HURR/   TIM(50) ,RCT(50) ,RAD(50) ,VRMAX(50) ,
1                  PZRO(50) ,TIMPOS(150),XPOS(150),YPOS(150)

C
COMMON /EXTRA/ X(45,15) ,Y(45,15)

C
DO 100 I=1, IM
DSDXI(I)= 0.
DO 100 J=1, JM
GRID1(I,J)=0.
GRID2(I,J)=0.
S(I,J)=0.
P(I,J)= 0.
X(I,J)= 0.
Y(I,J)= 0.
XX(I,J)= 0.
YY(I,J)= 0.
COSG(I,J)= 0.
SING(I,J)= 0.

```

```
100 CONTINUE
    NDEX= NMAX/INC+1
    DO 200 K=1,NDEX
        HYD1(K)=0.0
        HYD2(K)=0.0
        HYD3(K)=0.0
        HYD4(K)=0.0
        HYD5(K)=0.0
        HYD6(K)=0.0
        UCM1(K)=0.
        UCM2(K)=0.
        UCM3(K)=0.
        UCM4(K)=0.
        UCM5(K)=0.
        UCM6(K)=0.
        VCM1(K)=0.
        VCM2(K)=0.
        VCM3(K)=0.
        VCM4(K)=0.
        VCM5(K)=0.
        VCM6(K)=0.
200 CONTINUE
    IMAX = IM+1
    DO 300 I=1,IMAX
        DO 300 J=1,JM
            WIND(I,J) = 0.0
300 CONTINUE
    DO 400 J=1,JM
        DTDET(J)= 0.
400 CONTINUE
    DO 500 K=1,67
```

```
HOB$1(K)= 0.  
HOB$2(K)= 0.  
HOB$3(K)= 0.  
HOB$4(K)= 0.  
HOB$5(K)= 0.  
HOB$6(K)= 0.  
500 CONTINUE  
DO 600 K=1,50  
TIM(K)= 0.  
RCT(K)= 0.  
RAD(K)= 0.  
VRMAX(K)= 0.  
PZRO(K)= 0.  
600 CONTINUE  
DO 700 K=1,150  
TIMPOS(K)= 0.  
XPOS(K)= 0.  
YPOS(K)= 0.  
700 CONTINUE  
RETURN  
END  
SUBROUTINE FIELD  
COMMON/BLK1/GRID1(45,15),GRID2(45,15),  
10,HYD1(30),HYD2(300),HYD3(300),HYD4(300),UCM1(300),UCM2(300),UCM3(300),UCM  
24(300),VCM1(300),VCM2(300),VCM3(300),VCM4(300)  
COMMON/BLK2/C2,CC,C1XI,C1ETA,DXI,DETA,DELT,F,IM,JM,  
INM,JMM,N  
COMMON/BLK5/DSCX1(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6  
1(300),VCM5(300),VCM6(300)  
COMMON/BLK7/HOB$1(67),HOB$2(67),HOB$3(67),HOB$4(67),HOB$5(67),HOB$  
16(67)  
COMMON /TRANS/ XX(45,15) ,YY(45,15) ,CCSG(45,15),SING(45,15)
```

```
COMMON /HURR/  TIM(50) ,ROT(50) ,RAD(50) ,VRMAX(50)
1          PZRO(50) ,TIMPOS(150),XPOS(150),YPOS(150)
COMMON /HURRI/ YRANGE ,THIT      ,XHIT      ,YHIT      ,
3          C10       ,C11       ,PHI       ,
4          NT1       ,NT2       ,
5          PINF      ,IX        ,IY

C
11     READ(5,11) NT1
11     FFORMAT( 15 )
12     READ(5,12)(TIM(I),ROT(I),RAD(I),VRMAX(I),PZRC(I),I=1,NT1)
12     FFORMAT( 5F10.1)
13     READ(5,11) NT2
13     READ(5,13)(TIMPOS(I),XPOS(I),YPOS(I),I=1,NT2)
13     FORMAT(3F10.1)
13     WRITE(6,15)
15     FORMAT(1H1,15X,24HTHE STORM PARAMETERS ARE ,//,1X,
151H    TIME      ROTATION      RADIUS      MAX WINDS CEN PRESS ,/,1X,
250H    (HRS)      (DEG)      (NN)      (KNOTS)      (MB)      ,/)
16     WRITE(6,16)(TIM(I),ROT(I),RAD(I),VRMAX(I),PZRC(I),I=1,NT1)
16     FORMAT(1X,F8.0,F10.0,2F10.1,F10.0)
16     WRITE(6,17)
17     FORMAT(1H1,15X,37HTHE POSITIONS FOR THE STORM TRACK ARE ,//,1X,
129H    TIME      X COORD      Y COORD ,/,1X,
210H    (HRS)      ,/)
18     WRITE(6,18)(TIMPOS(I),XPOS(I),YPOS(I),I=1,NT2)
18     FORMAT(1X,F8.0,2F10.1)
NU1 = 5
DO 400 I=1,IM
READ(NU1,5) (GRID2(I,J),J=1,JM)
5 FORMAT(11F7.2)
400 CONTINUE
```

```
      DO 500 I=1,IM
      READ(NU1,6) (S(I,J),J=1,JM)
  6 FORMAT(5E14.7)
500 CONTINUE
      DO 600 I=1,IM
      READ(NU1,7) DSDXI(I)
  7 FORMAT(E14.7)
600 CONTINUE
      DO 700 J=1,JM
      READ(NU1,8) DTDET(J)
  8 FORMAT(2X,E14.7)
700 CONTINUE
      READ(NU1,9)(HCBS1(II),II=1,67)
      READ(NU1,9)(HOBS2(II),II=1,67)
      READ(NU1,9)(HCBS3(II),II=1,67)
      READ(NU1,9)(HCBS4(II),II=1,67)
      READ(NU1,9)(HOBS5(II),II=1,67)
      READ(NU1,9)(HCBS6(II),II=1,67)
  9 FORMAT(19F4.1)
      DO 300 II=1,67
      HOBS1(II)= 99999.
C      HOBS1(II)=HOBS1(II)*0.3048
C      HOBS2(II)=HOBS2(II)*0.3048
C      HOBS3(II)=HOBS3(II)*0.3048
C      HCBS4(II)=HOBS4(II)*0.3048
C      HCBS5(II)=HOBS5(II)*0.3048
C      HOBS6(II)=HOBS6(II)*0.3048
300 CONTINUE
      DO 101 I=1,IM
      READ (S,10) (XX(I,J),YY(I,J),J=1,JM)
 10 FORMAT (10F7.2)
```

```
101 CCNT INUE
    DC 102 I=1,IM
        READ(5,19) (COSG(I,J),SING(I,J),J=1,JM)
19    FORMAT (10F8.5)
102 CONTINUE
    DO 200 J=1,JM
        DC 100 I=1,IM
        IJ=I+J
        IF((IJ/2)*2.EQ.IJ)GO TO 110
        GRID2(I,J)=GRID2(I,J)*0.3048*6.0
        GO TO 100
110 CCNT INUE
        GRID1(I,J)=0.
        GRID2(I,J)=C.0
100 CCNT INUE
200 CCNTINUF
CCC*****
      WRITE(6,210)
210 FORMAT(1H1, //, 20X, 16HD E P T H S (N) , //, 7X, 5HS E A , 58X, 9HC D
      9 A S T , //)
      WRITE(6,211)(J,J=2,JMM,2)
211 FORMAT(8X,7(6X,2HJ=,I2),/)
      WRITE(6,212)(J,J=1,JM,2)
212 FORMAT(4X,8(6X,2HJ=,I2),//)
      DO 215 I=1,IM,2
      K= I+1
      WRITE(6,213) I,(GRID2(I,J),J=2,JMM,2)
213 FORMAT(1X,2HI=,I2,3X,7(4X,F6.1),/ )
      IF( I .EQ. IM ) GO TO 215
      WRITE(6,214) K,(GRID2(K,J),J=1,JM,2)
214 FORMAT(1X,2HI=,I2,8(4X,F6.1),/ )
```

```

215  CONTINUE
      RETURN
      END
      SUBROUTINE ELEV
      COMMON/BLK 1/GRID1(45,15),GRID2(45,15),
      S(45,15),HYD1(30
100),HYD2(300),HYD3(300),HYD4(300),UCM1(300),UCM2(300),UCM3(300),UCM
24(300),VCM1(300),VCM2(300),VCM3(300),VCM4(300)
      COMMON/BLK2/C2,CC,C1X,I,C1ETA,DX,I,DELT,F,IM,JM,
      IMM,JMM,N
      COMMON /BLK2C/ CCRIG,CS,C3,C4,RCT,IMN2,IMN3,JMM2
      COMMON/BLK5/DSXI(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6
1(300),VCM5(300),VCM6(300)
      COMMON/STCRM/NC,XEYE,YEYE,P(45,15),WIND(46,15)

C
      DO 180 J=3,JMN
      DTET=DTDET(J)*DETA
      IS= 4
      IMS= IMM3
      IF((J/2)*2.EQ.J) IS = 3
      IF(IS.EQ.3) IMS = IMM2
      JP1 = J+1
      JM1 = J-1
      DO 130 I=IS,IMS,2
      IP1 = I+1
      IM1 = I-1
      C9= DELT/S(I,J)/S(I,J)
      DSXI=DSXI(I)*DXI
      URIT = GRID1(IP1,J)*S(IP1,J)
      ULEF = GRID1(IM1,J)*S(IM1,J)
      VTOP = GRID2(I,JP1)*S(I,JP1)
      VBOT = GRID2(I,JM1)*S(I,JM1)
      GRID1(I,J) = GRID1(I,J)= C9*( (URIT=ULEF)/DSXI+(VTOP=VBOT)/DTET )

```

-60

```
130 CCNT INUE
180 CONTINUE
CC***  
DTET= DTDET(JM)*DETA
DO 170 II=2,IMM,2
IP1=II+1
IM1= II-1
C9= DELT/S(II,JM)/S(II,JM)
DSXI= DSDXI(II)*DXI
URIT= GRID1(IP1,JM)*S(IP1,JM)
ULEF= GRID1(IM1,JM)*S(IM1,JM)
IF( II .EQ. 2 .OR. II .EQ. IMM ) ULEF=URIT
VBOT= GRID2(II,JMM)*S(II,JMM)
VTOP= -VBOT
GRID1(II,JM)=GRID1(II,JM)=C9*((URIT-ULEF)/DSXI+(VTOP-VBOT)/DTET)
170 CONTINUE
C
DO 30 J=3,JMM2,2
II= 2
GRID1(II,J)=GRID1(II,J)=(DELT/S(II,J)/S(II,J))*(S(II,J+1)*GRID2(II
1,J+1)-S(II,J-1)*GRID2(II,J-1))/(DETA*DTDET(J))
II= IMM
GRID1(II,J)=GRID1(II,J)=(DELT/S(II,J)/S(II,J))*(S(II,J+1)*GRID2(II
1,J+1)-S(II,J-1)*GRID2(II,J-1))/(DETA*DTDET(J))
30 CONTINUE
CC***  
DO 150 II=2,IMM
IF((II/2)*2.EQ.II) GO TO 140
CCC GRID1(II,2)= P(II,2)= GRID2(II,3)/CS
GRID1(II,2)=0.25*(P(II-1,1)+GRID1(II-1,3)+P(II+1,1)+GRID1(II+1,3))
GO TO 150
```

```

140 BARO=P(II,1)
CCC GRID1(II,1) = BARO=GRID2(II,2)/CS
GRID1(II,1)= BARO
150 CONTINUE
RETURN
END
SUBROUTINE FLUX
COMMON/ELK1/GRID1(45,15),GRID2(45,15),
10,HYD1(30),HYD2(300),HYD3(300),HYC4(300),UCM1(300),UCM2(300),UCM3(300),UCM
24(300),VCM1(300),VCM2(300),VCM3(300),VCM4(300)
COMMON/ELK2/C2,CC,C1X,I,C1ETA,DX,I,DETA,DELT,F,IM,JM, IMM,JMM, N
COMMON /BLK20/ CORIO,CS,C3,C4,RCT,IMM2,IMM3,JMM2
COMMON/BLK5/DSDXI(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6
1(300),VCM5(300),VCM6(300)
COMMON/STORM/NO,XEYE,YEYE,P(45,15),WIND(46,15)

C
16
DO 8999 J=1,JM
DO 8999 I=1,IM,2
IP1=I+1
WABS=SQRT(WIND(I,J)**2+WIND(IP1,J)**2)
IF(WABS.LE.7.0)GO TO 9998
WCOF=1.100E-06+((1.-7./WABS)**2)*2.50E-06
GO TO 9997
9998 WCOF=1.100E-06
9997 DUWX=WIND(I,J)
DUWY=WIND(IP1,J)
WIND(I,J)=WCOF*WABS*DUWX
WIND(IP1,J)=WCOF*WABS*DUWY
8999 CONTINUE
C
DO 130 J=2,JMM

```

```
IS = 3
IMS = IMN2
IF((J/2)*2+EQ+J) IS=4
IF(IS.EQ.4)INS=IMN3
JP1 = J+1
JM1 = J-1
DO 80 I=IS,IMS,2
IP1 = I+1
IM1 = I-1
IWX=I
IWY=IP1
IF(IS=4)270,280,270
280 IWX=IM1
IWY=I
270 CONTINUE
CXI= C1XI/S(I,J)/DSDX(I)
CETA= C1ETA/S(I,J)/DTDET(J)
D1 = GRID1(IM1,J)=GRID2(IM1,J)
D2 = GRID1(IP1,J)=GRID2(IP1,J)
D3 = GRID1(I,JM1)=GRID2(I,JM1)
D4 = GRID1(I,JP1)=GRID2(I,JP1)
DELHX = GRID1(IP1,J)=GRID1(IM1,J)=(P(IP1,J)-P(IM1,J))
DELHY = GRID1(I,JP1)=GRID1(I,JM1)=(P(I,JP1)-P(I,JM1))
G = SQRT(GRID1(I,J)*GRID1(I,J)+GRID2(I,J)*GRID2(I,J))
AVHX = 0.5*(D1+D2)
AVHY = 0.5*(D3+D4)
DBAR = 0.5*(AVHY+AVHX)
G = (1.0+C2*F*Q/(DBAR*DBAR)).
G2 = G*G
A = GRID1(I,J)+RCT*GRID2(I,J)=CXI*AVHX*DELHX+C2*WIND(IWX,J)
B = GRID2(I,J)=ROT*GRID1(I,J)=CETA*AVHY*DELHY+C2*WIND(IWY,J)
```

8

```
      GRID1(I,J) = (G*A+C3*B)/(G2+C4)
      GRID2(I,J) = (G*B-C3*A)/(G2+C4)
 80 CONTINUE
 130 CONTINUE
CC*****
      DO 140 I=3,IMM2,2
      CXI=C1XI/(S(I,JM)*DSDXI(I))
      DELHX = GRID1(I+1,JM)-GRID1(I-1,JM)-(P(I+1,JM)-P(I-1,JM))
      Q = ABS(GRID1(I,JM))
      DBAR = 0.5*(GRID1(I-1,JM)-GRID2(I-1,JM)+GRID1(I+1,JM)-
      +GRID2(I+1,JM))
      G = (1.0+C2*F*Q/(DBAR*DBAR))
      GRID1(I,JM) = (GRID1(I,JM)-CXI*DBAR*DELHX+C2*WIND(I,JM))/G
 140 CONTINUE
      GRID1(1,JM)= GRID1(3,JM)
      GRID1(IM,JM)= GRID1(IMM2,JM)
CC*****
      DO 9000 II=2,IMM,IMM3
      DO 9000 J=3,JMM
      IF( (J/2)*2 .EQ. J ) GO TO 8100
      GO TO 8900
 8100 CCNTINUE
      CETA= C1ETA/S(II,J)/DTDET(J)
      DELHY=GRID1(II,J+1)-GRID1(II,J-1)-(P(II,J+1)-P(II,J-1))
      AVHY=0.5*(GRID1(II,J+1)-GRID2(II,J+1)+GRID1(II,J-1)-GRID2(II,J-1))
      Q= SQRT( GRID1(II,J)*GRID1(II,J)+GRID2(II,J)*GRID2(II,J) )
      G= (1.0+C2*F*Q /(AVHY*AVHY))
      III=3
      IF( III .EQ. IMM ) III= IMM2
      UBAR= 0.5*( GRID1(III,J+1)+GRID1(III,J-1) )
      UFLUX= UBAR + GRID1(II,J)
```

```
E= GRID2(II,J)=ROT*UFLUX=CETA*AVHY*DELHY+C2*WIND(II,J)
GRID2(II,J)=B/G
GRID1(II,J)=UBAR
GC TO 9000
8900 CONTINUE
    III= 1
    IIII= 3
    IF( II .EQ. IMM ) III= IM
    IF( II .EQ. IMM ) IIII= IMM2
    GRID1(III,J)= GRID1(IIII,J)
9000 CONTINUE
CC*****
CC SPECIAL CALCULATIONS FOR SEAWARD CORNERS
DO 9050 I=2,IMM,IMM3
    II= 3
    IF( I .EQ. IMM ) II= IMM2
    IS= II-1
    Q1= SQRT(GRID1(II,1)**2+GRID2(II,1)**2)
    Q2= SQRT(GRID1(I,2)**2+GRID2(I,2)**2)
    D1=(GRID1(II-1,1)-GRID2(II-1,1)+GRID1(II+1,1)-GRID2(II+1,1))/2.
    D2=(GRID1(I,1)-GRID2(I,1)+GRID1(I,3)-GRID2(I,3))/2.
    G1=(1.+C2*F*Q1/(D1**2))
    G2=(1.+C2*F*Q2/(D2**2))
    DELHX= GRID1(II+1,1)-GRID1(II-1,1)-(P(II+1,1)-P(II-1,1))
    DELHY= GRID1(I,3)-GRID1(I,1)-(P(I,3)-P(I,1))
    CXI= C1XI/S(II,1)/DSCXI(II)
    CETA= C1ETA/S(I,2)/DTDET(2)
    AB= GRID1(II,1)+C3*GRID2(II,1)=CXI*D1*DELHX+C2*WIND(II,1)
    AP= AF+.5*C3*GRID2(II+IS,2)
    BB= GRID2(I,2)-C3*GRID1(I,2)=CETA*D2*DELHY+C2*WIND(I,2)
    BP= BB-.5*C3*GRID1(II,3)
```

G91

```
DENOM = G1*G2+.25*C4
GRID1(II,1) = (AP*G2+BP*.5*C3)/DENOM
GRID2(I,2) = (BP*G1-AP*.5*C3)/DENOM
GRID1(I,2) = .5*(GRID1(II,3)+GRID1(II,1))
GRID2(II,1) = .5*(GRID2(I,2)+GRID2(II+IS,2))
9050 CONTINUE
GRID1(1,1) = GRID1(3,1)
GRID1(IM,1) = GRID1(IMM2,1)
RETURN
END
SUBROUTINE CRAW1
COMMON /BLK1/ GRID1(45,15),GRID2(45,15), S(45,15),HYD1(30
10),HYD2(300),HYD3(300),HYD4(300),UCM1(300),UCM2(300),UCM3(300),UCM
24(300),VCM1(300),VCM2(300),VCM3(300),VCM4(300)
COMMON /BLK2/ C2,CC,C1XI,C1ETA,DXI,DETA,DELT,F,IM,JM, IMM,JMM, N
COMMON /BLK20/ COR10,CS,C3,C4,ROT,IMM2,IMM3,JMM2
COMMON /BLK3/ GRAV, NMAX,IH1,JH1,IT1,JT1,IH2,
IH2,IT2,JT2,IH3,JH3,IT3,JT3,IH4,JH4,IT4,JT4,INC
COMMON /BLK5/ DSCX1(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6
1(300),VCM5(300),VCM6(300)
COMMON /BLK6/ IH5,JH5,IT5,JT5,IH6,JH6,IT6,JT6
C
DIMENSION UVEL(45,15),VVEL(45,15)
EQUIVALENCE (UVEL(2),VVEL)
C
MIN = ((N-1)*DELT)/60. + 1.E-5
IF(MOD(MIN,30).NE.0)RETURN
NH = N/INC + 1
HYD1(NH) = GRID1(IH1,JH1)
HYD2(NH) = GRID1(IH2,JH2)
HYD3(NH) = GRID1(IH3,JH3)
```

```
HYD4(NH) = GRID1(IH4,JH4)
HYD5(NH) = GRID1(IH5,JH5)
HYD6(NH) = GRID1(IH6,JH6)
CALL METER
C
C      IF (MOD(MIN,60).NE. 0) RETURN
C      CALCULATE THE X AND Y VELOCITIES
C          AT THE INTERIOR POINTS
DO 100 J=2,JMM
IS = 3
IMS = IMM2
IF ((J/2)*2.EQ. J) IS = 4
IF (IS .EQ. 4) INS = IMM3
CCC FOR RAD. LBC IS=2 AND IMS=IMM
CCC REM CHANGE IN DC 310
JP1 = J+1
JM1 = J-1
DO 100 I=IS,IMS,2
IP1 = I+1
IM1 = I-1
D1 = GRID1(IM1,J)=GRID2(IM1,J)
D2 = GRID1(IP1,J)=GRID2(IP1,J)
D3 = GRID1(I,JM1)=GRID2(I,JM1)
D4 = GRID1(I,JP1)=GRID2(I,JP1)
DBAR = .25*(D1+D2+D3+D4)
UVEL(I,J) = GRID1(I,J)/DBAR
VVEL(I,J) = GRID2(I,J)/DBAR
100 CONTINUE
C      BOUNDARY POINTS
DO 201 J=1,JM,JMM
DO 200 I=3,IMM2,2
```

```
IP1 = I+1
IM1 = I-1
D1 = GRID1(IM1,J)=GRID2(IM1,J)
D2 = GRID1(IP1,J)=GRID2(IP1,J)
DBAR = .5*(D1+D2)
UVEL(I,J) = GRID1(I,J)/DBAR
VVEL(I,J) = GRID2(I,J)/DBAR
200 CONTINUE
UVEL(IM,J) = GRID1(IM,J)/DBAR
VVEL(IM,J) = GRID2(IM,J)/DBAR
201 CONTINUE
I = 1
DO 202 J=1,JM,JMM
D1 = GRID1(2,J)=GRID2(2,J)
D2 = GRID1(4,J)=GRID2(4,J)
DBAR = .5*(D1+D2)
UVEL(I,J) = GRID1(I,J)/DBAR
VVEL(I,J) = GRID2(I,J)/DBAR
202 CONTINUE
DO 310 I=2,IMM,IMM3
DO 310 J=2,JMM
IF ((J/2)*2.EQ. J) GO TO 305
II = 1
IIII = 4
IF (I .EQ. IMM) II = IM
IF (I .EQ. IMM ) IIII = IMM3
D1= GRID1(I,J)=GRID2(I,J)
D2= GRID1(IIII,J)=GRID2(IIII,J)
DBAR= 0.5*(D1+D2)
UVEL(II,J) = GRID1(II,J)/DBAR
VVEL(II,J) = GRID2(II,J)/DBAR
```

```
      GO TO 310
305 CONTINUE
      JP1 = J+1
      JM1 = J-1
      D3 = GRID1(I,JP1)-GRID2(I,JP1)
      D4 = GRID1(I,JM1)-GRID2(I,JM1)
      DBAR= 0.5*(D3+D4)
      UVEL(I,J) = GRID1(I,J)/DBAR
      VVEL(I,J) = GRID2(I,J)/DBAR
310 CONTINUE
      DO 311 I=1,IM
      DO 311 J=1,JM
      IF( MOD(I+J,2) .EQ. 1 ) GO TO 311
      UVEL(I,J)= UVEL(I,J)*100.
      VVEL(I,J)= VVEL(I,J)*100.
311 CONTINUE
C
      T = MIN/60.
      WRITE (6,400) T
400 FCRMAT (1H1,6X,13HTIME (HRS) = ,F5.2,///,20X,12HE L E V (M),
1///,7X,5HS E A,58X,9HC D A S T,//)
      WRITE (6,405) (J,J=2,JMM,2)
405 FORMAT (8X,7(6X,2HJ=,I2),/ )
      - WRITE (6,406) (J,J=1,JM,2)
406 FFORMAT (4X,8(6X,2HJ=,I2),//)
      DO 415 I=1,IM,2
      K = I+1
      WRITE (6,410) I,(GRID1(I,J),J=2,JMM,2)
410 FORMAT (1X,2HI=,I2,3X,7(4X,F6.2),/ )
      IF (I .EQ. IM) GO TO 415
      WRITE (6,411) K,(GRID1(K,J),J=1,JM,2)
```

```
411 FORMAT (1X,2H I=,I2,8(4X,F6.2),/ )
415 CONTINUE
    WRITE (6,419) T
419 FORMAT (1H1,6X,13HTIME (HRS) = ,F6.2,///,20X,35HX AND Y V E L O C
1 I T I F S (CM/S),///,7X,5HS E A,E8X,9HC C A S T,//)
    WRITE (6,420) (J,J=2,JMM,2)
    WRITE (6,420) (J,J=1,JM,2)
DO 425 I=1,IM,2
K = I+1
    WRITE (6,421) I,(UVEL(I,J),J=1,JM,2)
    WRITE (6,422) (VVEL(I,J),J=1,JM,2)
421 FORMAT (1X,2H I=,I2,8(4X,F6.1))
422 FORMAT (5X,8(4X,F6.1),/ )
    IF (I .EQ. IM) GO TO 425
    WRITE (6,423) K,(UVEL(K,J),J=2,JMM,2)
    WRITE (6,424) (VVEL(K,J),J=2,JMM,2)
423 FORMAT (1X,2H I=,I2,3X,7(4X,F6.1))
424 FORMAT (8X,7(4X,F6.1),/ )
425 CONTINUE
    RETURN
END .
SUBROUTINE METER
COMMON/BLK1/GRIE1(45,15),GRID2(45,15),
10,HYD2(300),HYD3(300),HYD4(300),UCM1(300),UCM2(300),UCM3(300),UCM
24(300),VCM1(300),VCM2(300),VCM3(300),VCM4(300)
COMMON/BLK2/C2,CC,C1XI,C1ETA,CXI,DETA,DELT,F,IM,JM,   IMM,JMM, N
COMMON/BLK3/           GRAV,          NMAX,IH1,JH1,IT1,JT1,IH2,
1 JH2,IT2,JT2,IH3,JH3,IT3,JT3,IH4,JH4,IT4,JT4,INC
COMMON/BLK5/DSDXI(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6
1(300),VCM5(300),VCM6(300)
COMMON/BLK6/IH5,JH5,IT5,JT5,IH6,JH6,IT6,JT6
```

C

```
NUV= N/INC+1
DO 10 KK=1,6
  GO TO (20,30,40,50,60,70),KK
20 II = IT1
  JJ = JT1
  GO TO 80
30 II = IT2
  JJ = JT2
  GO TO 80
40 II = IT3
  JJ = JT3
  GO TO 80
50 II = IT4
  JJ = JT4
  GO TO 80
60 II=IT5
  JJ=JT5
  GO TO 80
70 II=IT6
  JJ=JT6
80 II = II+1
  I2 = II+1
  J1 = JJ+1
  J2 = JJ+1
  AVEDX = (GRID1(II,JJ)+GRID2(II,JJ)+GRID1(I2,JJ)+GRID2(I2,JJ))*0.5
  AVEDY = (GRID1(II,J1)+GRID2(II,J1)+GRID1(II,J2)+GRID2(II,J2))*0.5
  GO TO (110,120,130,140,150,160),KK
110 UCM1(NUV) = GRID1(IT1,JT1)/AVEDX
  VCM1(NUV) = GRID2(IT1,JT1)/AVEDY
  GO TO 10
```

120 UCM2(NUV) = GRID1(IT2,JT2)/AVEDX  
VCM2(NUV) = GRID2(IT2,JT2)/AVEDY  
GO TO 10  
130 UCM3(NUV) = GRID1(IT3,JT3)/AVEDX  
VCM3(NUV) = GRID2(IT3,JT3)/AVEDY  
GO TO 10  
140 UCM4(NUV) = GRID1(IT4,JT4)/AVEDX  
VCM4(NUV) = GRID2(IT4,JT4)/AVEDY  
GO TO 10  
150 UCM5(NUV)=GRID1(IT5,JT5)/AVEDX  
VCM5(NUV)=GRID2(IT5,JT5)/AVEDY  
GO TO 10  
160 UCM6(NUV)=GRID1(IT6,JT6)/AVEDX  
VCM6(NUV)=GRID2(IT6,JT6)/AVEDY  
10 CCNT INUE  
RETURN  
END  
SUBROUTINE FUV  
COMMON/BLK1/GRID1(45,15),GRID2(45,15),  
10,HYD1(30),HYD2(300),HYD3(300),HYD4(300),UCM1(300),UCM2(300),UCM3(300),UCM  
24(300),VCM1(300),VCM2(300),VCM3(300),VCM4(300)  
COMMON/BLK2/C2,CC,C1XI,C1ETA,DXI,DETA,DELT,F,IM,JM,  
1 IMM,JMM,N  
COMMON/BLK3/  
1 GRAV,  
1 NMAX,IH1,JH1,IT1,JT1,IH2,  
1 JH2,IT2,JT2,IH3,JH3,IT3,JT3,IH4,JH4,IT4,JT4,INC  
COMMON/BLK5/DSDXI(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6  
1(300),VCM5(300),VCM6(300)  
COMMON/BLK6/IH5,JH5,IT5,JT5,IH6,JH6,IT6,JT6  
COMMON/BLK7/HOBSS1(67),HOBSS2(67),HOBSS3(67),HOBSS4(67),HOBSS5(67),HOBSS  
16(67)  
C  
WRITE (6,505) IH1,JH1,IH2,JH2,IH3,JH3,IH4,JH4,IH5,JH5,IH6,JH6

```

      WRITE (6,506) IT1, JT1, IT2, JT2, IT3, JT3, IT4, JT4, IT5, JT5, IT6, JT6
505  FORMAT(1H1,41H ADJUSTED-COMPUTED HYDROGRAPH (M) AT ,6(I2,1H,,
6 I2,7X),/,30H AND THE OBSERVED WATER LEVEL ,// )
506  FORMAT(11X,34HSIMULATED CURRENT (M/S) AT ,6(I2,1H,,I2,7X),/
1//)
      J=1
      K = 1
      NDEX= NMAX/INC+1
      DO 500 I=1,NDEX
      T = (K-1)*DELT/3600.
      MIN= (K-1)*DELT/60. +1.0E-05
C   CORRECT COMPUTED H FOR INITIAL OBSERVED WATER LEVEL CONDITIONS
      HYD1(I)= HYD1(I)+1.
C   HYD1(I)= HYD1(I)+HOBS1(1)
      HYD2(I)= HYD2(I)+HOBS2(1)
      HYD3(I)= HYD3(I)+HOBS3(1)
      HYD4(I)= HYD4(I)+HOBS4(1)
      HYD5(I)= HYD5(I)+HOBS5(1)
      HYD6(I)= HYD6(I)+HOBS6(1)
      WRITE (6,510) K,T,HYD1(I),HYD2(I),HYD3(I),HYD4(I),HYD5(I),HYD6(I)
      IF( MOD(MIN,60).EQ.0)WRITE(6,513)HOBS1(J),HOBS2(J),HOBS3(J),HOBS4(
9J),HOBS5(J),HOBS6(J)
      WRITE (6,511) UCM1(I),UCM2(I),UCM3(I),UCM4(I),UCM5(I),UCM6(I)
      WRITE (6,512) VCM1(I),VCM2(I),VCM3(I),VCM4(I),VCM5(I),VCM6(I)
      K = K+INC
      IF(MOD(MIN,60).EQ.0) J=J+1
500  CONTINUE
510  FORMAT (/,13X,2HN=,I5,10H T (HRS)=,F6.2,5H + ,6(F6.2,6X))
511  FORMAT (39X,4HU ,6(F7.2,5X))
512  FORMAT (39X,4HV ,6(F7.2,5X))
513  FORMAT( 34X,7HOBS H ,6(F6.2,6X))

```

RETURN  
END  
SUBROUTINE WINDF  
COMMON /BLK2/C2,CC,C1XI,C1ETA,DXI,DETA,DELT,F,IM,JM, IMM,JMM, N  
COMMON / TRANS/XX(45,15),YY(45,15),COSG(45,15),SING(45,15)  
COMMON /HURR/ TIM(50),ROT(50),RAD(50),VRMAX(50),  
1 PZERO(50),TIMPOS(150),XPOS(150),YPOS(150)  
COMMON /HURR1/ YRANGE,THIT,XHIT,YHIT,  
3 C10,C11,PHI,  
4 NT1,NT2,  
5 PINF,IX,IY  
COMMON /EXTRA/ X(45,15),Y(45,15)  
COMMON / STCRM /NO,XEYE,YEYE,P(45,15),WIND(46,15)  
C  
DATA IT1P1,IT2P1/2,2/  
C  
NC=N  
TIME=FLOAT(N-1)\*DELT/3600.  
DO 101 I=IT1P1,NT1  
IF(TIM(I).GE.TIME) GO TO 102  
101 CONTINUE  
I=NT1  
102 IT1=I-1  
DO 103 I=IT2P1,NT2  
IF(TIMPCS(I).GE.TIME) GO TO 104  
103 CONTINUE  
I=NT2  
104 IT2= I-1  
IT1P1=IT1+1  
IT2P1=IT2+1  
TFAC1=(TIME-TIM(IT1))/(TIM(IT1P1)-TIM(IT1))

```
TFAC2=(TIME-TIMPOS(IT2))/(TIMPOS(IT2P1)-TIMPOS(IT2))
DELT2=(TIMPOS(IT2F1)-TIMPOS(IT2))*3600.

C
R=RAD(IT1)+TFAC1*(RAD(IT1P1)-RAD(IT1))
VR=VRMAX(IT1)+TFAC1*(VRMAX(IT1P1)-VRMAX(IT1))
THETA=ROT(IT1)+TFAC1*(ROT(IT1P1)-ROT(IT1))
PO=PZRC(IT1)+TFAC1*(PZRO(IT1P1)-PZRO(IT1))
XEYE=XPOS(IT2)+TFAC2*(XPOS(IT2P1)-XPOS(IT2))
YEYE=YPOS(IT2)+TFAC2*(YPOS(IT2P1)-YPOS(IT2))
US=(XPOS(IT2F1)-XPOS(IT2))/DELT2
VS=(YPOS(IT2P1)-YPOS(IT2))/DELT2
R=R*.84167
VR=VR*.515
US=US*2201.6
VS=VS*2201.6
THETA=(90.-THETA+PHI)*.0174533
CO=COS(-THETA)
CI=SIN(-THETA)

C
USR=US*CO + VS*CI
VSR=VS*CO - US*CI

C
DO 108 I=1,IX
DO 107 J=1,IY
XP=XX(I,J)
YP=YY(I,J)
XF1=XP-XEYE
YP1=YP-YEYE
RSML=SQRT(XP1**2 + YP1**2)
IF(MOD((I+J),2).EQ.1) GO TO 109
A=-YP1*C10 -XP1*C11
```

```
B= XPI*C10 - YPI*C11
IF(RSML.LT.1.0E+10) RSML= 1.0E+10
IF(RSML.LE. R) GO TO 105
C
C POINT IS OUTSIDE RADIUS TO MAX WINDS
C
S1=R/(R+RSML)
S2=(VR/RSML)*(R/RSML)**.45
GO TO 106
C
C POINT IS INSIDE RADIUS TO MAX WINDS
C
105 S1=RSML/(R+RSML)
S2=(VR/RSML)*((RSML/R)**1.5)
106 X1=(USR*S1)+(A*S2)
Y1=(VSR*S1)+(B*S2)
C
C ROTATE VELOCITY FOR CURV GRID
C
C=COSG(I,J)
S=SING(I,J)
XR=X1*C+Y1*S
YR=Y1*C-X1*S
X(I,J)=XR
Y(I,J)=YR
GO TO 107
109 RDIST=22.*.84167
PP=PO+(FINF=PO)*EXP(-RDIST/RSML)
P(I,J)=(FINF=FF)*.01
107 CONTINUE
108 CCNTINUE
```

```
DO 110 J=1,IY
DO 110 I=1,IX,2
IF((J/2)*2.EQ.J)GO TO 112
WIND(I,J)=X(I,J)
WIND(I+1,J)=Y(I,J)
GO TO 110
112 IF(I.EQ.IX)GO TO 110
WIND(I,J)=X(I+1,J)
WIND(I+1,J)=Y(I+1,J)
110 CCNTINUE
RETURN
END
FUNCTION YLAND(XD)
DATA YRANGE/17./
YLAND=ABS(XD-YRANGE)/7.
RETURN
END
```