

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

GB
450
.T43
NW 76-3
v. 2

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.



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	
7. AUTHOR(s) John J. Wanstrath, Robert E. Whitaker, Robert O. Reid, and Andrew C. Vastano		6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Texas A&M Research Foundation F.E. Box H College Station, Texas 77843		8. CONTRACT OR GRANT NUMBER(s) DACW72-73-C-0014	
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Army Coastal Engineering Research Center Kingman Building, Fort Belvoir, Virginia 22060		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS A31231	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE November 1976	
		13. NUMBER OF PAGES Vol. I, 166 Vol. II, 176	
		15. SECURITY CLASS. (of this report) UNCLASSIFIED	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
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)			
<p>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.</p>			

(Continued)

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).

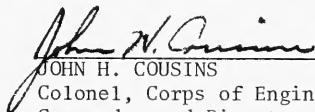
This volume was prepared by John J. Wanstrath; Volume I was prepared by John J. Wanstrath, Robert E. Whitaker, Robert O. Reid, and Andrew C. Vastano, Department of Oceanography, Texas A&M University, College Station, Texas, under CERC Contract No. DACW72-73-C-0014. Most of the computational work in the 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 ¹

¹ 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 ξ -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 ξ and η .
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</u> arclength.
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</u> arclength.
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, $0 < \xi < \lambda$, = 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, JMAS1, IL, VARWT	Continuation flag, number of shore- 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 re-started, the following cards must be supplied:

(4) Card 4

B, BZRO The values of β 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

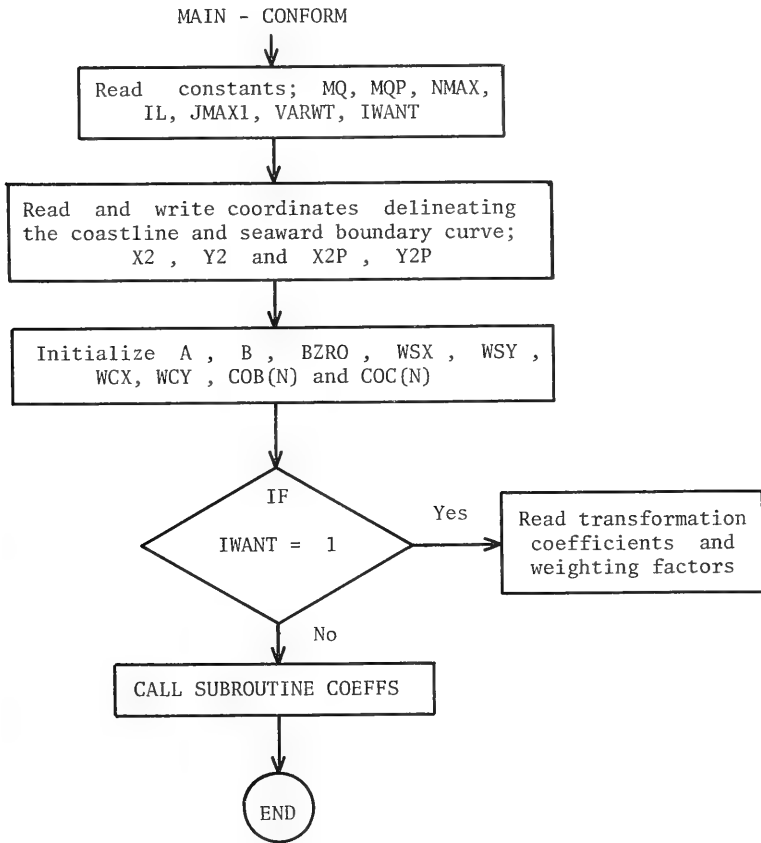
WSX,WSY The value of the weighting factors for the seaward boundary curve (x and y component) and WCX,WCY coastline (x and y component) from the last iteration of the previous run (4E14.7).

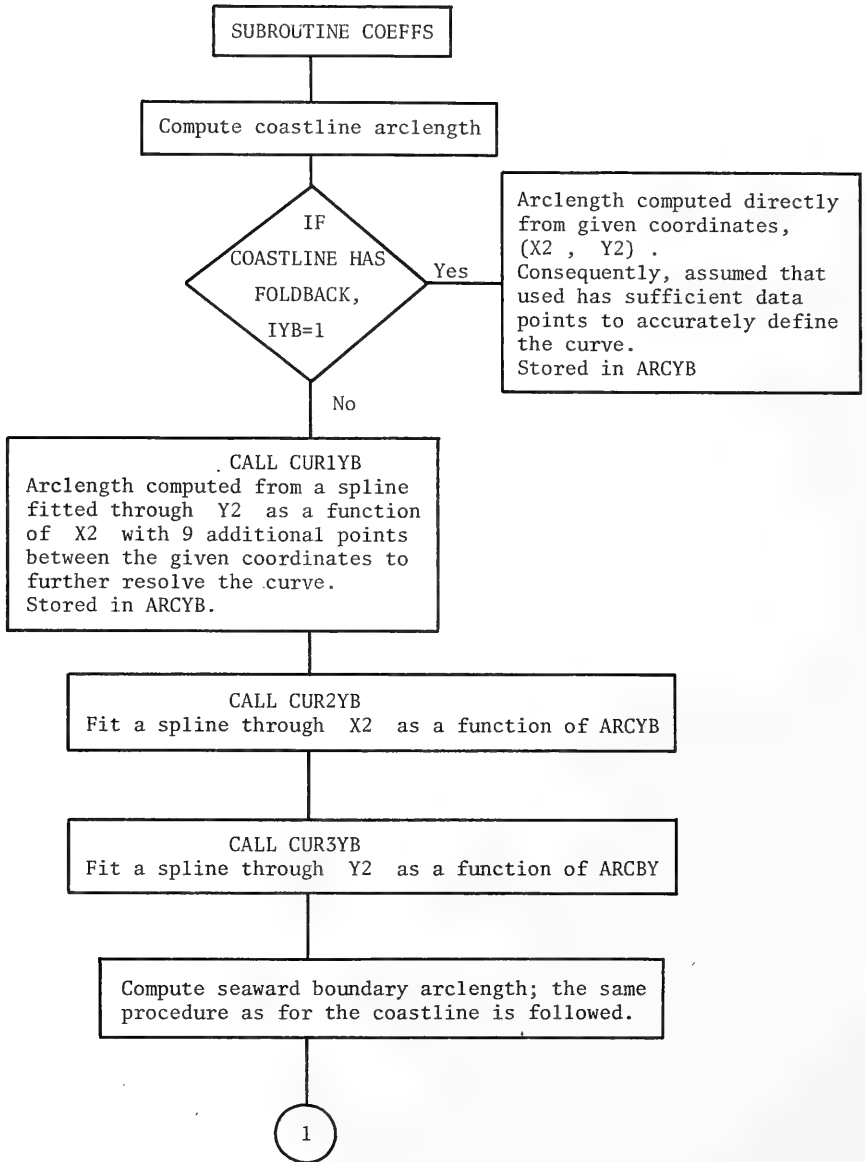
5. Mathematical Procedures and Program Limitations.

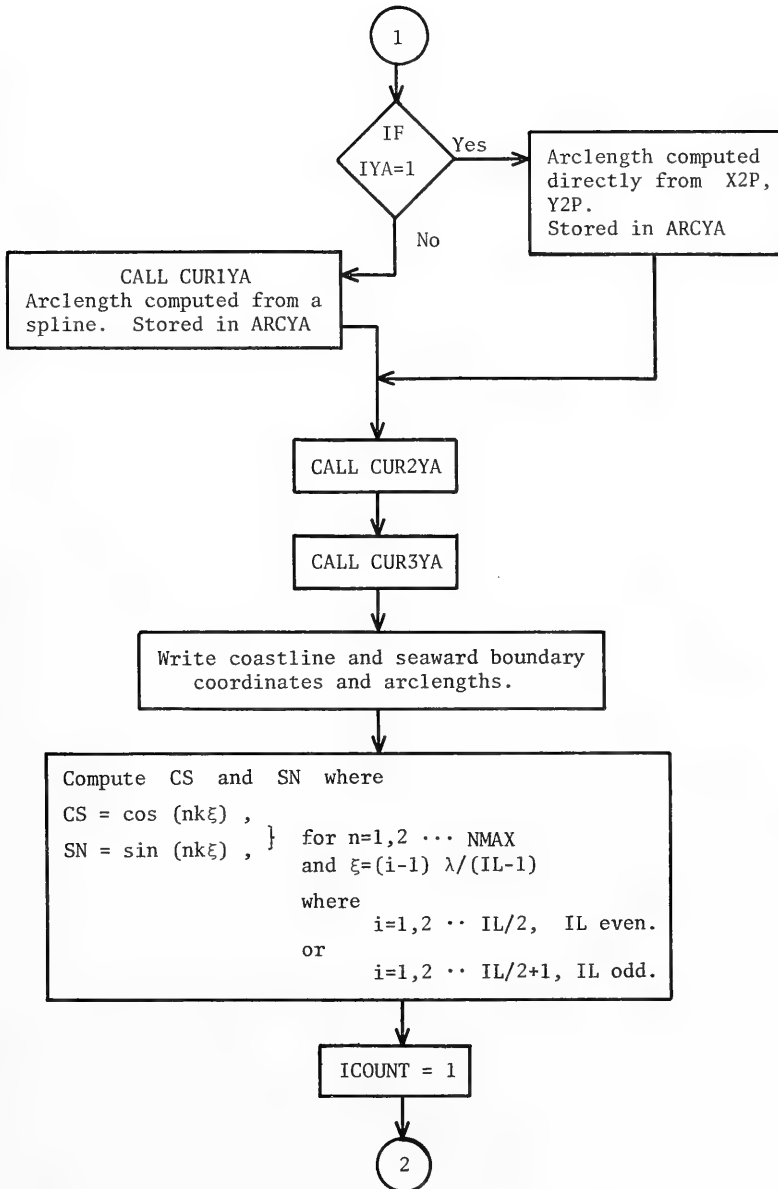
The conformal mapping relations, solutions for the transformation coefficients, and the iterative procedure for determining the coefficients are presented in Volume I of this report. The mapping equations are sufficiently general to treat the situation where either or

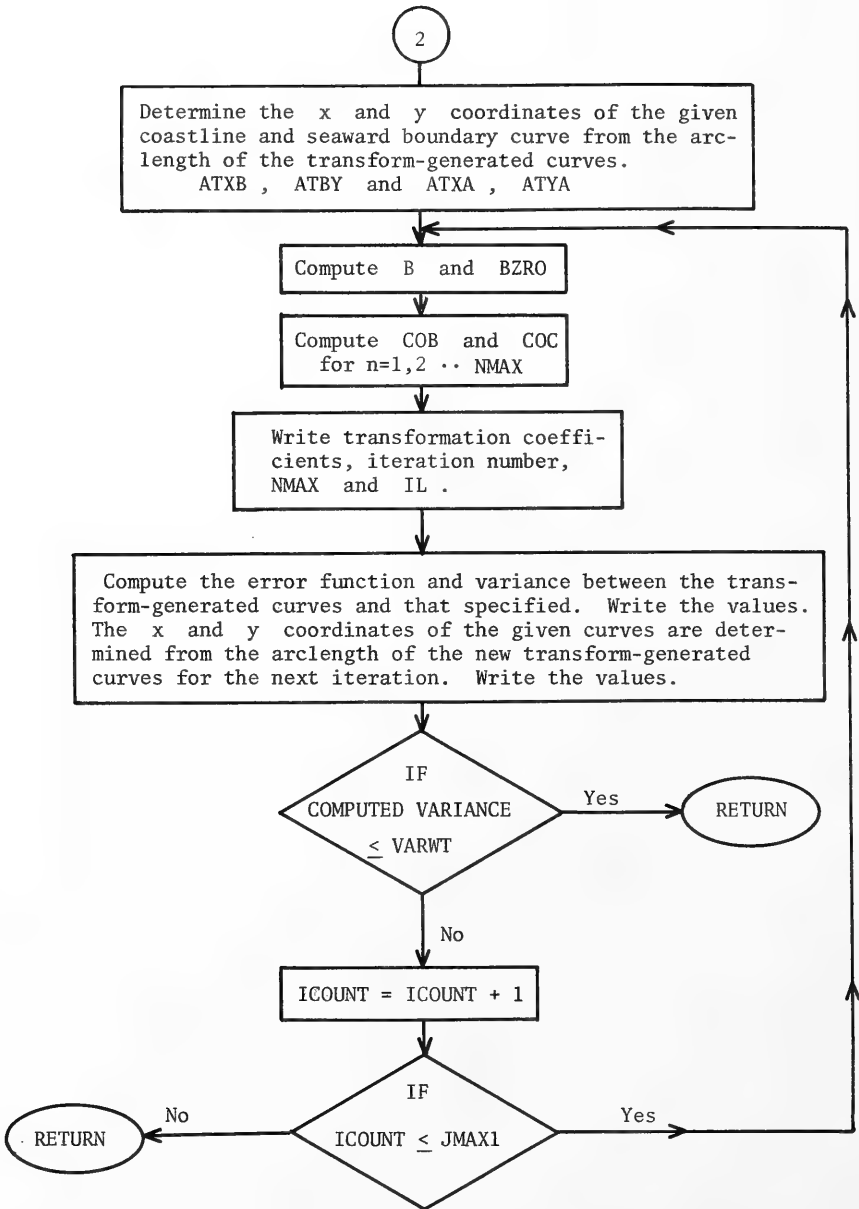
both given curves are multivalued in y for a specified x value. The only program limitation relates to the computer memory capacity. No program error messages or consistency checks are incorporated in this routine.

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 ξ (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 β

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 (←) on the right-hand side of the page.

The results from CONFORM required by Program GRID are β , B_n , and the coefficients, B_n and C_n . 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
C

I N P U T D A T A

1 47 40 150 1500*001

360.00 110.00

C
C
C
C
C

COASTLINE COORDINATES

C
C
C
C
C

SEAWARD ECUNDARY COORDINATES

000.00 C35.00
005.60 C40.00
010.50 C44.80
018.00 C51.00
028.00 C59.00
040.00 C66.80
050.50 C73.00
056.50 C76.50
063.00 C82.00
070.00 C53.00
074.00 102.00
076.50 108.50
079.00 116.00
082.50 127.00
088.00 137.80
095.00 146.00
106.00 154.00
121.00 162.00
136.00 167.00
153.00 168.30
158.00 168.00
161.50 167.30
166.00 168.80
171.00 169.00
183.00 168.00
197.00 166.00
214.00 165.00
230.00 166.00
242.00 167.00
250.00 166.70
253.40 167.20
257.00 169.50
265.00 169.00
280.00 166.00
291.00 163.00
296.00 159.30
298.50 159.90
303.00 159.00
314.00 163.00
323.00 148.00
328.00 142.00
331.00 137.20
335.50 129.20
342.50 121.00
349.50 115.00
352.50 113.00

000.00 002.00
007.50 005.00
016.00 010.00
021.50 014.50
028.00 019.00
036.00 023.00
045.00 028.00
054.00 032.00
060.00 036.00
066.00 040.00
077.00 046.00
085.00 051.00
089.00 055.00
095.00 064.00
104.00 076.00
110.00 084.00
112.00 089.00
113.50 095.00
114.50 101.00
117.50 108.00
121.00 112.00
127.00 116.00
139.00 121.00
153.00 123.50
165.00 123.00
179.00 119.00
193.00 115.00
213.00 104.00
229.00 092.00
244.00 080.00
257.00 071.00
267.00 066.00
276.00 062.00
289.00 056.00
298.00 049.00
307.00 042.00
318.00 035.00
333.00 027.00
347.00 020.00
360.00 014.00

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 ξ -axis to the x -axis .

The program GRID is composed of

MAIN	Defines constants. Reads transformation coefficients outputted 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 (μ , ν , 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 $\theta = \tan^{-1} \left(\frac{\partial y / \partial \xi}{\partial x / \partial \xi} \right)$.
SUBROUTINE CURV9	Contains the expansion curve $Y = Z + B(X^C)$ where A , B , and C are constants. The term Y is either S_p (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 ξ for determining the transform-generated coastline arclength as a function of ξ ;
- c) 51 evenly spaced values of η for determining the arclength along a particular isoline of ξ as a function of η ;
- d) the computing grid of 47 ξ (or S*) lines and 15 η (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 ξ isolines and 2 evenly spaced increments between η 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 ξ
NUMETA, DELSS,	lines + 2, number of η lines, ΔS^* in nautical
SSTRT, DELTT,	miles, first value of S^* in nautical miles,
ND, NS	ΔT^* in minutes, number of depths, number of
	points in format 314, 3F5.1, 214.

(2) Card 2

BETA BZRO	The value of β 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 traveltime coordinate T.
----	---

5. Mathematical Procedures and Program Limitations.

Information concerning the expanding grid procedure and the relations transforming ξ, η to S^*, T^* space is presented in Volume I of this report. The user is required to know the coefficients 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 ΔS^* 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 nautical miles $\leq S^* \leq$ 236 nautical miles
(10 intervals of ΔS^*)

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

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

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

For region II, 236 nautical miles $\leq S^* \leq$ 260 nautical miles
(4 intervals of ΔS^*)

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

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

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

For region III, 260 nautical miles $\leq S^* \leq$ 302 nautical miles
(7 intervals of ΔS^*)

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

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

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

For region IV, 302 nautical miles $\leq S^* \leq$ 356 nautical miles
(9 intervals of ΔS^*)

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

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

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

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

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

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

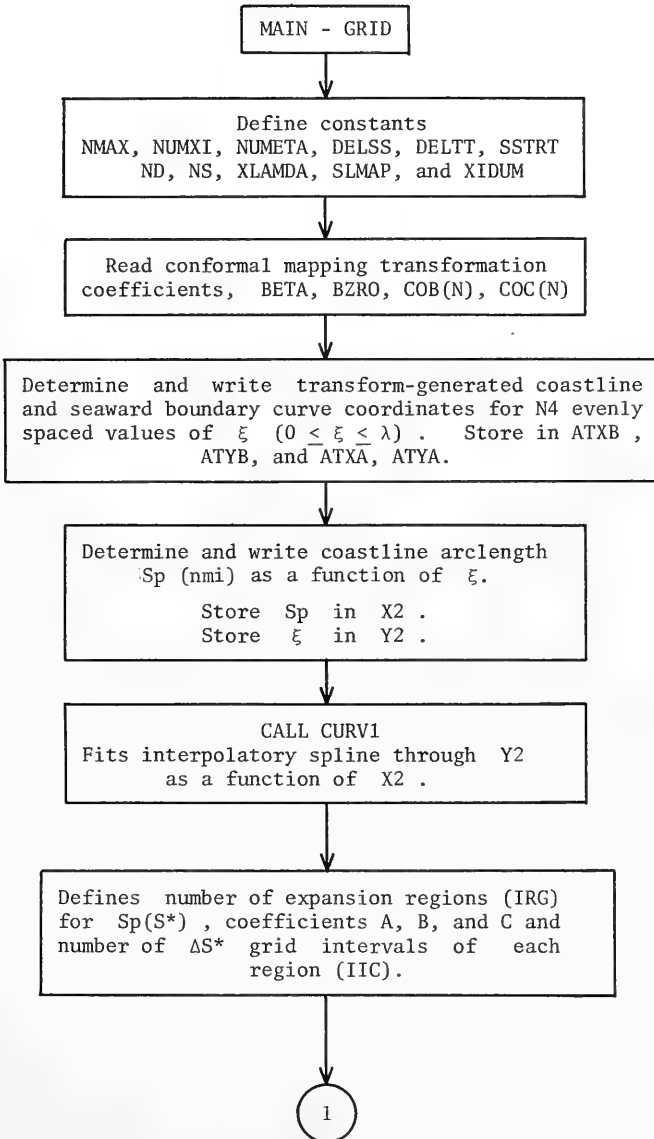
$$S_p = 356 \text{ nautical miles at } S^* = 356 \text{ nautical miles}$$

A similar procedure is followed for the expansion function

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

where T is the long wave traveltime (minutes) along the particular isoline of ξ . 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 $\Delta T^* =$ total long wave traveltime-number of η grid intervals), the program computes ΔT^* .

6. Flow Chart.



1

Determine appropriate values of ξ for evenly spaced values of S^* .

For $i = 1, 2 \dots \text{NUMXI}$

$$S_i^* = \text{SSSTRT} + (i-1) \cdot \text{DELSS}$$

CALL CURV9
Computes S_p for a given S^* .

FUNCTION CURV2
Returns a value for ξ given S_p .
Arrays X2 and Y2 cannot be altered from the last call of CURV1.

Store ξ in XI.

Determine and write coordinates and arclength S_n (nmi) along a particular isoline of ξ (=XIDUM) for NS evenly spaced values of η ,
 $-\beta \leq \eta \leq \beta$.

Store S_n in X2.

Store η in Y2.

CALL CURV 1

2

2

Read water depth (fathoms) at ND evenly spaced positions from the seaward boundary curve to the coastline.

Determine and write the long wave traveltime T (min) and distance S_n (nmi) along XIDUM for the ND positions.

Store T in X2P.

Store S_n in Y2P.

CALL CURV3

Fits interpolatory spline through Y2P as a function of X2P.

Define number of expansion regions (RG) for $T(T^*)$, coefficients A, B, and C, and number of ΔT^* grid intervals of each region (IIC). If $IIC(1) = NUMETA-1$, compute ΔT^* .

3

3

Determine appropriate values of η
for evenly spaced values of T^* .

For $j = 1, 2 \dots \text{NUMETA}$

$$T_j^* = (j-1) \times \text{DELTT}$$

CALL CURV9

Computes T for a given T^*

Function CURV4

Returns a value for S_n given
 T . Arrays $X2P$ and $Y2P$ can-
not be altered from the last
call of CURV3.

Function CURV2

Returns a value for η given
 S_n . Arrays $X2$ and $Y2$ can-
not be altered from the last
call of CURV1.

Store η in ETA.

The values of ξ and η are now known.
Determine the computing grid data.

4

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 .. NUMXI-1
j = 1,2 .. NUMETA.

Determine and write scale factor μ (units
x,y space-nmi).

Store μ in SX for i=1,2 .. NUMXI-2.

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

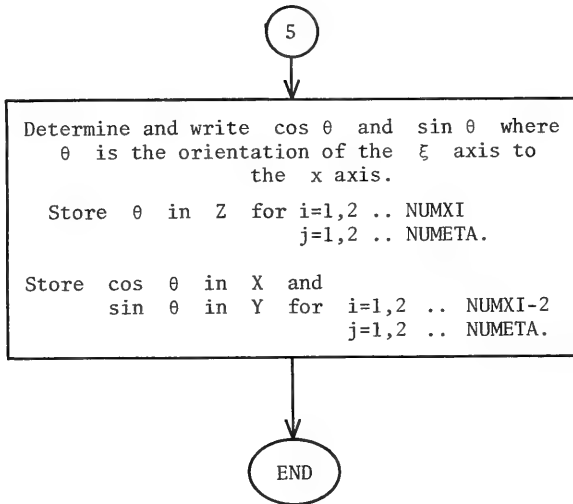
Store ν in SY for j=1,2 .. NUMETA.

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

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

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

5



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 \times 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 ξ for determining the computing grid data.
ETA	Dimensioned NUMETA. Contains the values of η for determining the computing grid data.

Constants

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

- DELSS The value in nautical miles of ΔS^* . 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 μ values are dimensionless, i.e.,
- $$DXI = 1852 \frac{m}{nmi} \times \Delta S^* \text{ nmi} \times \left[SLMAP \frac{nmi}{x,y \text{ unit}} \times \mu \text{ unit} \right]$$
- DELTT The value in minutes at ΔT^* . This corresponds to DETA in Program SSURGE.
- Since in SSURGE, the product, $\nu \Delta T^*$, is always computed in the surge equations, we determine DETA in meters such that ν values are dimensionless, i.e.,
- $$DETA = 1852 \frac{m}{nmi} \times \Delta T^* \text{ min} \times \left[SLMAP \frac{nmi}{x,y \text{ unit}} \times \nu \text{ unit} \right]$$
- 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 traveltime.
- NS The number of points along XIDUM for determining η 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 ξ 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⁻²).
- N4 The number of points used in determining ξ 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 sub-
and divided into IQUAD intervals between ξ lines and
JQUAD JQUAD intervals between η lines.

8. Input and Output.

The first card input to Program GRID gives the number of conformal transformation coefficients NMAX, the number of ξ lines NUMXI, the number of η lines NUMETA, ΔS^* in nautical miles DELSS, the first value of S^* in nautical miles SSSTRT, ΔT^* 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 β 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 ξ -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, $\partial Sp / \partial S^*$, S^* , Sp , and ξ are listed. Additionally, the transform-generated arclengths at values of η for evenly spaced increments of T^* and at η values for constant increments of T along the chosen ξ -line, the η values along the specified ξ -line such that ΔT^* is constant, and the traveltime and depths along the constant ξ -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 μ and ν related to the transformation of ξ to S^* and η to T^* , respectively, the scale factor F , and the sines and cosines of theta giving the orientation of the ξ -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.

C
C
INPUT DATA
150 47 15 6C176.0 8.0 31 51
C

2.1109529EG01 9.5078648E01
-4.3493873EG01 -6.8176833E01
-3.1662011EG01 1.5427266E01
1.2422513EG01 -1.1718220E00
-5.06365C2EG00 -1.4470718E-02
1.6771864EG00 -1.9216334E00
-3.5976884EG00 -6.7330628E-01
1.4192860EG00 1.3667185E00
-1.1794944EG00 -3.0913591E-01
4.1190876E-01 1.3357793E-01
-3.7759354E-01 4.8326302E-03
7.9716016E-02 5.3212195E-02
-1.0800307E-01 1.6050344E-01
-1.2900039E-02 -1.556875E-01
-5.6365976E-02 1.4022560E-01
-1.3795311E-02 -6.3873108E-02
-3.5084819E-02 9.6181509E-02
6.6049823E-03 -4.1274305E-02
-1.4301171E-02 2.1339555E-02
9.3927601E-03 -2.2866277E-02
-1.1320465E-02 2.3924600E-02
-1.0776871E-03 -2.2690670E-03
-1.2749315E-02 6.3207555E-03
4.0667667E-03 1.4115754E-03
-1.6674068E-03 4.6394670E-04
3.9959578E-04 -2.6365667E-04
-1.8662185E-03 5.6828282E-04
9.0772555E-04 7.4038904E-04
-1.5591983E-03 -4.9429248E-04
5.1903310E-04 3.3489054E-04
1.1438246E-03 -1.0357530E-04
3.7952317E-04 1.7815754E-04
-6.7125324E-04 9.4876230E-05
4.3712874E-04 -5.7567649E-06
-5.8352881E-04 1.4706647E-04
1.8351595E-04 -3.2129854E-05
-2.5129644E-04 3.0412098E-04
-1.0435642E-05 -2.4449992E-04
1.0222874E-05 2.3265936E-04
-8.8439003E-06 -1.9700862E-04
-1.2788127E-05 1.5418616E-04
-2.6134267E-05 -9.3255390E-05
-1.4402552E-05 1.0334477E-04
-5.0798231E-06 -5.2775377E-05
-2.3834472E-05 3.2466107E-05
2.6851103E-05 -2.6723854E-06
-4.3115519E-05 -2.0363452E-06
2.6351927E-05 1.1422653E-05
-2.3699234E-05 2.3080753E-06
1.0604752E-05 -2.8166598E-06
1.1909960E-05 5.3967162E-06
7.9651199E-06 -7.3427355E-07
-9.4350325E-06 3.0565014E-06
3.5921445E-06 -2.8028600E-06
-3.1637865E-06 5.2027666E-06
8.0697328E-07 -3.2049353E-06
-3.1561970E-06 2.9766227E-06

2.0821353E-06 -3.1002452E-07
-2.7802158E-06 7.3753132E-07
1.13094917E-06 -7.1666702E-07
-6.1223962E-07 1.4500429E-06
-1.8451712E-07 -1.1197107E-06
-3.3654205E-07 1.1016565E-06
8.9384915E-08 -4.5424002E-07
-3.0120346E-07 5.2354470E-07
-3.6734064E-08 -3.5825709E-07
-1.4140805E-07 4.2440565E-07
7.3883689E-08 -1.2894558E-07
-3.2262466E-07 5.8860751E-08
2.7314736E-07 1.0023264E-07
-2.8025695E-07 -3.2327644E-08
1.2341097E-07 1.6892800E-08
-9.4987154E-08 8.2818246E-08
1.9075643E-08 -6.2256573E-08
-3.8559006E-08 7.6237824E-08
1.2275724E-08 -4.7234358E-08
-2.2807855E-08 5.9522317E-08
-4.7725860E-09 -3.6176467E-08
-1.3059312E-08 4.0413841E-08
1.3466543E-08 -1.2140226E-08
-3.1784570E-08 -4.018408E-11
2.6290277E-08 1.2313708E-08
-2.8336831E-08 -5.5426238E-09
1.3901584E-08 4.2926331E-09
-1.0413045E-08 3.6285344E-09
2.9425597E-09 -3.9004828E-09
-2.4943488E-09 7.0901456E-09
-2.1406580E-09 -6.0237420E-09
1.8588775E-09 8.3549490E-09
-3.7924576E-09 -6.9248194E-09
1.5027985E-09 6.0568206E-09
-9.6309918E-10 -2.5732890E-09
-1.5450322E-09 1.53651158E-09
1.7949878E-09 3.7305856E-10
-2.4545629E-09 -5.5504161E-10
1.9894453E-09 5.3920530E-10
-2.0184501E-09 -6.3272567E-10
1.3540433E-09 7.49506038E-10
-1.1897698E-09 -2.6100463E-10
5.6070780E-10 5.2285846E-11
-2.8726818E-10 3.0456696E-10
-9.8993701E-11 -3.7664508E-10
9.5877297E-11 5.1835164E-10
-1.9201876E-10 -3.8510567E-10
8.7366526E-11 1.4544234E-10
-8.5212951E-11 -2.1783821E-10
-3.6849378E-12 1.8260446E-10
-3.1853177E-12 -8.2957043E-11
-5.8541698E-11 6.1865899E-11
5.4400211E-11 -7.0616809E-12
-6.423174E-11 3.3844000E-12
5.4853889E-11 1.7602039E-11
-5.9319842E-11 -1.1763089E-12
3.5082522E-11 7.7555677E-12
-3.3595148E-11 3.4384777E-12
1.3693600E-11 3.8907021E-13
-2.0626758E-11 5.8511549E-12
1.0770427E-11 -1.6112032E-12
-1.0974893E-11 6.7705536E-12

3.7490653E-12 -5.0353240E-12
-3.6685470E-12 8.1811602E-12
-4.5895235E-13 -4.9445672E-12
-7.4799452E-13 7.4302943E-12
-1.2979236E-12 -5.1512560E-12
9.3319544E-14 5.3514749E-12
-8.5127448E-13 -3.4656560E-12
-4.6265112E-13 2.2230644E-12
2.7429158E-13 -1.3656647E-12
-1.5414227E-12 2.7822358E-13
1.6202911E-12 2.6397256E-13
-1.9462274E-12 -2.8659264E-13
1.4287833E-12 6.2960244E-13
-1.4235914E-12 -1.8242733E-13
8.0951983E-13 2.4162215E-13
-6.7456456E-13 1.1457485E-13
2.6952090E-13 -1.5477038E-13
-1.6877584E-13 4.0572501E-13
9.9264509E-14 -3.7385145E-13
6.9077629E-14 4.7001651E-13
-1.2943800E-13 -2.406812E-13
1.7026387E-14 2.5116468E-13
1.2898778E-15 -1.2329172E-13
-1.0197886E-13 -6.332572E-14
9.0036274E-14 -6.3540502E-15
-1.2530695E-13 -1.0823757E-15
1.0618124E-13 4.4320389E-14
-1.1324192E-13 -1.3624101E-14
7.0773638E-14 2.4508613E-14
-6.2031906E-14 3.0256047E-15
3.4432727E-14 -1.1037802E-14
-1.9752800E-14 3.7600872E-14
95
63
53
46
39
32
30
28
28
26
25
23
22
21
20
18
17
16
15
14
13
12
11
11
10
9
7
6
5
2

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 ξ -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 15).
- (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 (ξ,η) 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, μ , transforming ξ 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 v transforming $\bar{\eta}$ 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 (IH_k, JH_k) 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, 1-1,2..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 θ and sine θ where θ is the angle between the ξ 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.

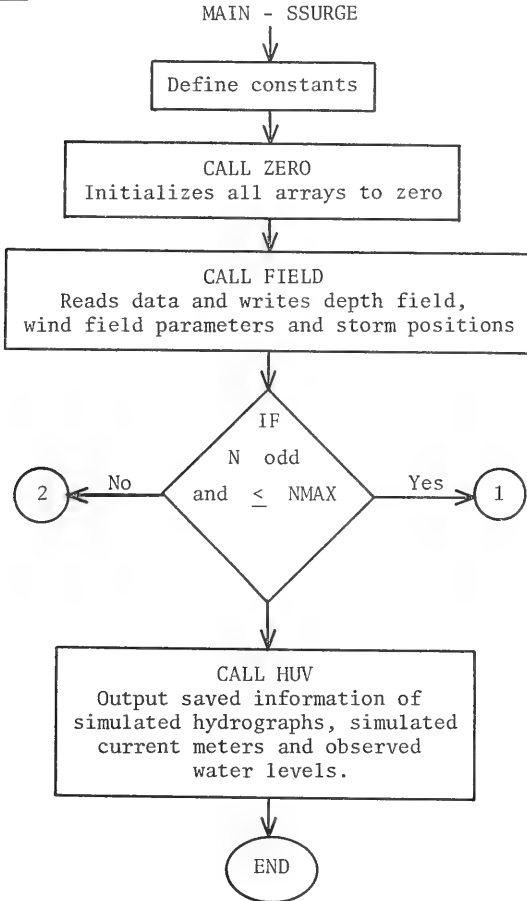
5. Mathematical Procedures and Program Limitations.

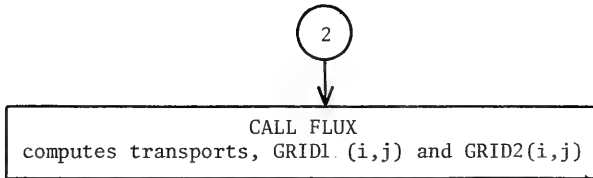
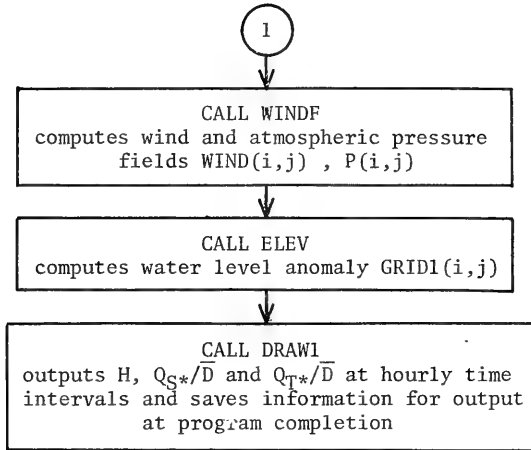
The storm surge equations, solutions, and algorithms are described in the text of Volume I of this report.

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_0 , 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 (ξ,η) space.
DSDXI	Dimensioned IM . Contains the dimensionless scale factor, μ , transforming ξ to S^* .
DTDET	Dimensioned JM . Contains dimensionless scale factor, ν , transforming η 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 θ relating the orientation of the ξ -axis to the x-axis at the computational grid points.
SING	Dimensioned $IM \times JM$. Contains sine value of θ relating the orientation of the ξ -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 $i+j$.
HOBSk k=1,2,..6	Each array is dimensioned 67. Contains the observed hourly water levels in meters at grid locations (I_{Hk} , J_{Hk}).

HYDK Each array is dimensioned 300. Contains values
 k=1,2..6 of the computed water level H in meters at
 grid locations (IHK, JHK).

UCMK Each array is dimensioned 300. Contains values of
 k=1,2..6 the vertically averaged water velocity, Q_{S^*}/\bar{D} , in
 meters per second at grid location (ITk, JTk).

VCMk Each array is dimensioned 300. Contains values
 k=1,2..6 of the vertically averaged water velocity, Q_{T^*}/\bar{D} ,
 in meters per second at grid location (ITk, JTk).

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 HYDK, UCMK and VCMK. 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).

IHK, JHK
k=1,2..6 Grid point location for saving the computed water level. The sum of the indexes must be odd. Index IHK cannot exceed IM. Index JHK cannot exceed JM.

ITK, JTK
k=1,2..6 Grid point location for saving the vertically averaged water velocities. The sum of the indexes must be even. Index ITK cannot exceed IM. Index JTK cannot exceed JM-1.

CORIO Coriolis parameter ($=6.70875 \times 10^5$ second⁻¹) 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_1 and K_2 , are 1.1×10^{-6} and 2.5×10^{-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 μ) is read along the specified row and the scale factor array DTDET (an alias for ν) 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
C
C
C
C
C
C
C
C
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.

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.	27.	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.	185.6	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

50.	158.9	164.8								
51.	160.1	174.1								
52.	162.7	183.5								
53.	161.0	190.6								
54.	160.0	195.5								
55.	158.8	200.1								
56.	158.6	204.5								
57.	159.3	213.4								
58.	160.4	220.9								
59.	167.0	232.2								
60.	170.3	239.9								
66.	193.8	303.7								
110.00	32.50	26.50	24.00	23.00	22.00	21.00	17.00	15.00	13.00	11.00
9.00	9.00	7.00	2.30							
95.00	37.00	27.00	24.00	23.50	21.70	21.00	18.50	16.00	12.00	10.00
9.00	8.00	7.00	2.30							
100.00	52.00	39.00	35.00	26.00	23.00	21.50	19.50	17.00	16.00	13.00
10.20	8.00	6.50	2.30							
100.00	54.00	44.00	35.00	32.00	26.50	23.50	21.00	18.50	16.00	14.00
12.00	9.00	7.70	2.30							
100.00	57.00	46.00	38.00	33.00	29.00	25.00	22.00	19.00	17.00	14.40
12.50	9.00	7.70	2.00							
100.00	57.00	41.00	37.00	33.00	28.50	24.00	22.00	18.00	16.00	14.00
12.50	9.00	7.00	2.00							
105.00	58.00	46.50	38.00	33.00	28.00	25.00	21.50	18.50	16.00	14.00
12.50	9.50	6.50	1.70							
105.00	58.00	47.00	38.00	32.00	29.00	25.00	22.00	18.00	16.00	13.00
12.00	9.50	7.00	1.50							
105.00	57.00	46.00	40.00	32.00	28.00	25.50	21.00	17.50	15.00	13.50
11.70	8.00	6.50	1.50							
100.00	59.00	48.50	37.00	33.00	27.00	24.00	20.00	17.00	15.00	12.50

10.70	8.50	6.00	1.50							
108.00	60.00	47.00	36.00	33.00	28.00	23.50	19.00	16.50	14.00	12.00
10.50	8.00	6.00	1.50							
108.00	61.00	47.00	37.50	32.00	27.00	22.00	19.00	16.00	14.00	12.00
10.00	8.00	6.00	1.50							
102.00	61.00	36.00	36.00	31.00	27.00	22.00	17.50	15.70	13.00	11.70
9.50	8.00	6.00	1.50							
108.00	60.00	40.00	35.00	30.00	25.00	21.50	18.00	15.00	13.50	11.50
9.00	7.70	5.50	1.30							
110.00	55.00	35.00	33.00	26.50	22.50	20.00	16.00	14.00	13.00	11.50
11.00	7.70	5.50	1.30							
102.00	36.00	31.00	27.50	24.00	21.00	17.50	15.00	13.70	12.70	11.50
11.00	7.70	5.50	1.30							
110.00	35.00	31.00	25.00	22.50	20.00	17.00	14.00	12.50	12.00	11.50
9.50	8.00	6.00	1.30							
115.00	34.50	29.00	25.00	21.50	18.00	16.00	13.70	13.00	11.50	11.00
9.00	8.00	6.00	1.30							
100.00	34.00	28.00	24.00	21.50	18.00	15.00	13.50	12.50	11.50	10.50
8.50	7.50	5.50	1.30							
100.00	34.00	27.00	24.00	21.00	17.50	15.50	14.00	12.00	11.50	11.00
8.70	7.00	5.50	1.30							
100.00	35.00	27.00	24.00	20.50	16.50	15.00	14.50	14.00	11.00	10.00
8.50	7.00	5.50	1.30							
100.00	36.00	27.00	24.00	20.50	17.50	15.70	15.70	14.00	11.00	10.00
8.70	7.50	5.50	1.20							
102.00	34.00	26.50	22.00	19.00	16.00	16.00	15.00	13.00	10.50	10.00
8.00	7.30	5.50	1.20							
100.00	35.00	27.00	23.00	18.50	18.00	16.00	15.00	14.00	10.50	10.00
9.00	7.30	5.00	1.20							
100.00	36.00	27.00	22.00	19.00	18.50	16.00	15.00	13.50	12.00	10.00
7.00	6.20	4.00	1.00							

100.00	30.00	26.70	21.00	21.00	18.00	16.00	15.00	13.00	11.50	11.00
10.00	7.00	4.00	1.20							
95.00	32.00	26.00	21.00	20.50	17.00	15.00	14.20	12.00	11.00	11.00
9.00	8.00	5.00	1.20							
100.00	31.00	26.00	21.00	19.00	17.00	15.00	14.00	11.50	10.50	10.50
9.00	8.00	5.50	1.20							
95.00	31.00	24.50	21.50	19.00	17.00	15.00	12.50	11.00	9.00	10.50
9.00	7.70	5.70	1.20							
100.00	31.50	24.00	21.50	18.50	16.00	14.50	11.50	11.00	9.00	9.50
9.00	7.50	5.70	1.20							
90.00	31.00	24.00	21.50	18.50	15.00	13.00	10.00	11.00	9.00	9.00
9.00	7.50	5.40	1.20							
90.00	31.00	25.00	21.00	18.00	14.00	13.00	10.00	10.50	9.00	9.00
9.00	7.50	5.00	1.20							
90.00	32.00	26.00	21.00	18.00	15.50	14.00	11.00	8.50	8.70	8.70
9.50	7.00	5.00	1.20							
100.00	33.00	26.50	21.50	17.50	15.00	13.50	11.00	6.50	8.50	8.50
8.00	7.00	5.20	1.20							
105.00	37.00	27.00	21.50	17.00	14.00	13.00	11.00	7.50	7.50	7.50
8.00	7.00	5.30	1.20							
95.00	37.50	28.00	20.50	16.00	13.50	12.50	10.50	9.00	8.50	7.20
7.20	6.70	5.30	1.20							
95.00	37.00	28.00	19.50	14.50	13.00	12.50	11.00	10.00	8.00	7.00
7.00	6.30	5.00	1.20							
95.00	36.00	27.50	18.00	18.00	12.00	12.00	10.50	9.00	8.00	7.20
5.00	6.30	4.00	1.00							
90.00	35.50	26.00	19.00	14.50	11.50	11.50	11.50	8.50	7.50	7.50
5.00	6.00	4.00	1.00							
95.00	34.00	26.00	19.00	15.00	11.00	12.00	11.00	8.70	8.00	7.00
5.00	6.00	4.20	1.00							
110.00	34.00	25.50	19.00	14.70	13.00	13.00	11.00	9.00	9.00	6.00

6.50	5.50	3.50	1.00															
90.00	33.00	24.00	18.00	14.00	13.70	13.50	10.00	9.00	8.50	7.50								
6.50	5.30	4.00	1.00															
100.00	30.00	24.00	18.50	14.50	12.00	12.50	10.00	10.00	8.70	8.00								
7.00	5.50	3.50	1.00															
95.00	32.50	23.00	17.50	16.00	14.00	12.50	11.50	9.50	9.00	8.00								
7.00	5.00	3.00	1.00															
100.00	32.00	23.00	18.00	14.50	12.00	9.00	7.00	4.50	4.00	4.50								
4.50	3.50	2.00	1.00															
1.0172672E+00	9.9516271E-01	9.4466743E-01	9.4466743E-01	5.1536421E-01	8.9332403E-01													
8.7534267E-01	8.5939932E-01	8.4452597E-01	8.3051112E-01	8.1688751E-01														
8.0388299E-01	7.9152409E-01	7.6050849E-01	7.7135815E-01	7.6675171E-01														
1.0097754E+00	9.8739227E-01	9.4298173E-01	9.2921875E-01	9.2188848E-01														
9.1690692E-01	9.1275140E-01	9.0877529E-01	9.0468989E-01	9.0017266E-01														
8.9513610E-01	8.8950278E-01	8.8367037E-01	8.7838425E-01	8.7564104E-01														
8.5317548E-01	8.6591520E-01	8.9643738E-01	9.1856354E-01	9.3645691E-01														
9.5152151E-01	9.6486679E-01	9.7736410E-01	9.8873806E-01	9.9931363E-01														
1.0087875E+00	1.0170814E+00	1.0238449E+00	1.0292237E+00	1.0320270E+00														
7.2862247E-01	7.6937524E-01	8.5680077E-01	9.1015655E-01	9.5152551E-01														
9.8653922E-01	1.0188401E+00	1.0502062E+00	1.0814706E+00	1.1135835E+00														
1.1464152E+00	1.1800811E+00	1.2125262E+00	1.2411963E+00	1.2582532E+00														
7.0555727E-01	7.5130990E-01	8.6154956E-01	9.1781872E-01	9.7150079E-01														
1.0184812E+00	1.0631359E+00	1.1079567E+00	1.1537737E+00	1.2026690E+00														
1.2546128E+00	1.310274E+00	1.3655835E+00	1.4167106E+00	1.4442715E+00														
7.1886750E-01	7.6234315E-01	8.6330465E-01	9.3281046E-01	9.9077934E-01														
1.0423636E+00	1.0918313E+00	1.1416816E+00	1.1926742E+00	1.2470169E+00														
1.3046266E+00	1.3660631E+00	1.4279816E+00	1.4860104E+00	1.5178702E+00														
7.5191737E-01	7.9198295E-01	8.8226825E-01	9.5003427E-01	1.0077327E+00														
1.0592898E+00	1.1084268E+00	1.1572291E+00	1.2060331E+00	1.2564063E+00														
1.3076091E+00	1.3585046E+00	1.4088730E+00	1.4522794E+00	1.4751297E+00														
7.8391955E-01	8.18890461E-01	9.0216528E-01	9.6672656E-01	1.0223101E+00														

1.0718998E+00 1.1185774E+00 1.1642956E+00 1.2086944E+00 1.2527648E+00
1.2951806E+00 1.3351412E+00 1.3696783E+00 1.3964246E+00 1.4091448E+00
8.0663957E-01 8.3351565E-01 9.1965284E-01 9.8150601E-01 1.0348947E+00
1.0824769E+00 1.1271904E+00 1.1705289E+00 1.2123601E+00 1.2534428E+00
1.2924935E+00 1.3288346E+00 1.3600063E+00 1.3843633E+00 1.3961778E+00
8.2190674E-01 8.5498656E-01 9.3361515E-01 9.5359494E-01 1.0452593E+00
1.0913936E+00 1.1349669E+00 1.1775737E+00 1.2192853E+00 1.2611334E+00
1.3021733E+00 1.342082E+00 1.3780654E+00 1.4082439E+00 1.4236955E+00
8.3429629E-01 8.6706595E-01 9.4456653E-01 1.002514E+00 1.0525469E+00
1.0976812E+00 1.1402067E+00 1.1823122E+00 1.2243469E+00 1.2677951E+00
1.3122516E+00 1.3578526E+00 1.4021456E+00 1.4420959E+00 1.4635727E+00
8.4577210E-01 8.7790270E-01 9.5369717E-01 1.0101340E+00 1.0580234E+00
1.1005600E+00 1.1408989E+00 1.1809084E+00 1.2211007E+00 1.2631291E+00
1.3069084E+00 1.3529727E+00 1.3990561E+00 1.4421836E+00 1.4659172E+00
8.5530650E-01 8.8737342E-01 9.6224702E-01 1.0164148E+00 1.0613176E+00
1.1003705E+00 1.1366526E+00 1.1720153E+00 1.2068525E+00 1.2425949E+00
1.2790253E+00 1.3161828E+00 1.3515242E+00 1.3822413E+00 1.3982396E+00
8.6351731E-01 8.9642320E-01 9.7173647E-01 1.0235274E+00 1.0649517E+00
1.0997273E+00 1.1308098E+00 1.1596621E+00 1.1865648E+00 1.2125062E+00
1.2376313E+00 1.2632404E+00 1.2900735E+00 1.3185774E+00 1.3368578E+00
8.7703029E-01 9.1138207E-01 9.8776624E-01 1.0366029E+00 1.0740952E+00
1.1043808E+00 1.1302224E+00 1.1526608E+00 1.1714372E+00 1.1862511E+00
1.1956295E+00 1.1984269E+00 1.1946972E+00 1.1885670E+00 1.1855618E+00
8.9672269E-01 9.2106568E-01 1.0061433E+00 1.0520015E+00 1.0865709E+00
1.1144722E+00 1.1381757E+00 1.1589232E+00 1.1766976E+00 1.1915770E+00
1.2027448E+00 1.2101712E+00 1.2157329E+00 1.2263343E+00 1.2364615E+00
9.4110785E-01 9.7074823E-01 1.0362568E+00 1.0768704E+00 1.1086509E+00
1.1335937E+00 1.1562894E+00 1.1777028E+00 1.1966724E+00 1.205335E+00
1.2461544E+00 1.2774106E+00 1.3172433E+00 1.3676201E+00 1.4010683E+00
9.9267286E-01 1.0142342E+00 1.0647794E+00 1.1000176E+00 1.1278735E+00
1.1510485E+00 1.1718939E+00 1.1918198E+00 1.2114881E+00 1.2321059E+00

1.2538956E+00 1.2767999E+00 1.2982941E+00 1.3145365E+00 1.3223745E+00
1.0472334E+00 1.0605520E+00 1.0957587E+00 1.1257822E+00 1.1500264E+00
1.1700470E+00 1.1876593E+00 1.2040417E+00 1.2194658E+00 1.2346651E+00
1.2495921E+00 1.2643925E+00 1.2784237E+00 1.2911920E+00 1.2982324E+00
1.0845379E+00 1.0953034E+00 1.1251858E+00 1.1521844E+00 1.1734529E+00
1.1902778E+00 1.2042667E+00 1.2162591E+00 1.2264973E+00 1.2353772E+00
1.2428660E+00 1.2491881E+00 1.2544189E+00 1.2590602E+00 1.2617635E+00
1.0997966E+00 1.1155237E+00 1.1534301E+00 1.1802228E+00 1.1996320E+00
1.2137925E+00 1.2244176E+00 1.2322633E+00 1.2375140E+00 1.2402913E+00
1.2404725E+00 1.2380778E+00 1.2335389E+00 1.2281342E+00 1.2250069E+00
1.069648E+00 1.1311535E+00 1.1834348E+00 1.2120158E+00 1.2306863E+00
1.2430269E+00 1.2510958E+00 1.2557001E+00 1.2570693E+00 1.2552271E+00
1.250161E+00 1.242001E+00 1.2316527E+00 1.2208096E+00 1.2146432E+00
1.1247198E+00 1.1554492E+00 1.2191683E+00 1.2500388E+00 1.2687797E+00
1.2802178E+00 1.2867819E+00 1.2893883E+00 1.2883591E+00 1.2837446E+00
1.2757348E+00 1.2647055E+00 1.2515245E+00 1.2390173E+00 1.2316532E+00
1.1627618E+00 1.1955711E+00 1.2634889E+00 1.2956086E+00 1.3153959E+00
1.3268146E+00 1.3329675E+00 1.3349355E+00 1.3329921E+00 1.3272115E+00
1.3178158E+00 1.3051658E+00 1.2904383E+00 1.2747581E+00 1.2652761E+00
1.2173315E+00 1.2492947E+00 1.3164416E+00 1.3500210E+00 1.3702622E+00
1.3824999E+00 1.3894357E+00 1.3920759E+00 1.3907543E+00 1.3854671E+00
1.3763197E+00 1.3635360E+00 1.3481916E+00 1.3315270E+00 1.3213927E+00
1.2816222E+00 1.3116924E+00 1.3761275E+00 1.4102892E+00 1.4318191E+00
1.4456532E+00 1.4544230E+00 1.4591288E+00 1.4600536E+00 1.4571876E+00
1.4504831E+00 1.4400886E+00 1.4274125E+00 1.4157200E+00 1.4100780E+00
1.3515534E+00 1.3795064E+00 1.4407199E+00 1.4753685E+00 1.4985526E+00
1.5146752E+00 1.5262421E+00 1.5242630E+00 1.5395012E+00 1.5406130E+00
1.5389180E+00 1.5338518E+00 1.5261394E+00 1.5186611E+00 1.5152652E+00
1.4211356E+00 1.4478293E+00 1.5070245E+00 1.5421295E+00 1.5672308E+00
1.5861296E+00 1.6011596E+00 1.6133369E+00 1.6229676E+00 1.6305292E+00
1.6361950E+00 1.6401801E+00 1.6415276E+00 1.6373639E+00 1.6320713E+00

1.49202367E+00 1.5151256E+00 1.5713631E+00 1.6058148E+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
1.8650425E+00 1.8873117E+00 1.6391338E+00 1.6747159E+00 1.7039928E+00
1.7293187E+00 1.7524575E+00 1.7741182E+00 1.7940665E+00 1.8125184E+00
1.8393711E+00 1.8468454E+00 1.8694602E+00 1.9021134E+00 1.9255364E+00
1.6418245E+00 1.6616685E+00 1.7090335E+00 1.7442923E+00 1.7758022E+00
1.8049900E+00 1.8322815E+00 1.8611925E+00 1.8879352E+00 1.9125255E+00
1.9311228E+00 1.9387464E+00 1.9270075E+00 1.9004238E+00 1.8814047E+00
1.7225043E+00 1.7381167E+00 1.778070E+00 1.8121023E+00 1.8456544E+00
1.8788385E+00 1.9126375E+00 1.9483185E+00 1.9846672E+00 2.0215143E+00
2.0548005E+00 2.0774502E+00 2.0778233E+00 2.0607204E+00 2.0493568E+00
1.7965964E+00 1.8076521E+00 1.8395907E+00 1.8723475E+00 1.9074805E+00
1.9442498E+00 1.9836687E+00 2.0268065E+00 2.0735916E+00 2.1271386E+00
2.1880780E+00 2.2634694E+00 2.3688770E+00 2.5493790E+00 2.6917623E+00
1.8778135E+00 1.8982734E+00 1.9022305E+00 1.9324185E+00 1.9686607E+00
2.0087526E+00 2.0534395E+00 2.1041572E+00 2.1618982E+00 2.2310803E+00
2.3171132E+00 2.4346587E+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 2.4501811E+00 2.6070034E+00 2.6550618E+00 2.7371290E+00
2.0412583E+00 2.0277659E+00 2.0125785E+00 2.0352485E+00 2.0715286E+00
2.1155270E+00 2.1166865E+00 2.2268750E+00 2.2948935E+00 2.3746489E+00
2.4651689E+00 2.5640250E+00 2.6562304E+00 2.7158523E+00 2.7427186E+00
2.1231956E+00 2.0988750E+00 2.0627685E+00 2.0808518E+00 2.1161562E+00
2.1628188E+00 2.2137515E+00 2.2756525E+00 2.3459652E+00 2.4267147E+00
2.5173933E+00 2.6222969E+00 2.7561489E+00 2.9517423E+00 3.0957613E+00
2.2101971E+00 2.1725416E+00 2.1117769E+00 2.1248165E+00 2.1579439E+00
2.2018436E+00 2.2255072E+00 2.3179535E+00 2.3854715E+00 2.4700442E+00
2.5544966E+00 2.6348629E+00 2.6984390E+00 2.7643277E+00 2.8146395E+00
2.2531337E+00 2.2209190E+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 2.7485051E+00 2.8323611E+00 2.8952547E+00
 2.2515640E+00 2.2155313E+00 2.1567544E+00 2.1658727E+00 2.1919805E+00
 2.2290327E+00 2.2769025E+00 2.3373294E+00 2.4115608E+00 2.5041475E+00
 2.168129E+00 2.2752964E+00 2.3098486E+00 3.0523688E+00 3.2103936E+00
 2.1934112E+00 2.1798798E+00 2.1593699E+00 2.167969CE+00 2.1882922E+00
 2.2163509E+00 2.2590232E+00 2.3127557E+00 2.3820815E+00 2.4738014E+00
 2.5936655E+00 2.7491925E+00 2.9336425E+00 3.1266695E+00 3.2373314E+00
 2.1029439E+00 2.1228976E+00 2.1560113E+00 2.1604552E+00 2.1688814E+00
 2.1847413E+00 2.205903E+00 2.2456691E+00 2.2956448E+00 2.3658014E+00
 2.4632818E+00 2.6074555E+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 2.1212615E+00 2.1383993E+00
 2.1649236E+00 2.2025968E+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 1.885666CE+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
 1.4645743E+00 1.3918863E+00 1.3282498E+00 1.2792876E+00 1.2565270E+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
3.432000E-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
5.5128417E-01 42

6.1	6.7	6.6	6.8	7.0	6.8	6.5	6.6	6.7	7.0	7.7	7.6	8.2	8.6	8.7	8.3	8.6	8.9	8.8	PP2
8.5	8.2	8.2	8.5	8.8	8.6	8.1	8.1	8.1	8.1	8.0	8.0	8.6	8.5	7.8	8.0	7.9	7.3	7.2	PP3
7.4	6.6	6.6	5.9	6.0	6.4	6.3	6.2	6.2	5.7	5.2								PP4	
2.2	2.3	2.4	2.5	2.8	2.9	3.1	3.1	3.1	2.9	2.7	2.6	2.6	2.4	2.3	2.4	2.5	2.6	2.6	MB1
3.0	3.7	4.3	4.6	5.0	5.4	5.6	5.7	5.9	6.1	6.3	6.7	6.9	7.1	7.2	7.2	7.3	7.3	7.5	MB2
7.7	8.0	8.2	8.4	8.6	8.7	8.9	8.7	8.5	8.2	8.2	8.2	8.4	8.6	8.6	8.6	8.6	8.6	8.4	MB3
8.2	8.0	8.0	8.0	8.0	8.1	8.1	8.2	8.2	8.2	8.2	8.1	8.2	8.2	8.2	8.2				MB4
3.4	3.3	3.3	3.3	3.4	3.6	3.1	2.8	3.0	3.2	3.2	3.2	3.9	4.2	4.7	4.7	5.0	5.0	5.3	SP1
5.4	5.4	5.5	5.3	5.4	5.8	5.6	5.7	6.0	6.1	6.4	6.2	6.2	6.2	6.1	6.1	6.0	6.0	6.0	SP2
6.5	6.9	7.4	7.3	7.2	6.9	7.0	6.8	6.5	6.4	5.8	6.5	6.7	6.7	6.8	6.7	6.4	6.1	6.7	SP3
5.4	5.7	5.6	5.5	5.7	5.8	5.8	6.2	6.2	5.5	5.8	5.3	4.8	4.4	4.0	4.1				SP4
103.01	74.99	95.97	80.17	91.56	83.09	88.30	85.18	85.89	86.74										
83.71	88.02	81.80	89.16	80.02	90.20	78.37	91.14	76.85	91.99										
75.35	92.79	73.93	93.53	72.65	54.17	71.63	54.66	70.50	95.00										
112.05	89.59	103.36	92.98	98.57	95.32	95.06	97.12	92.38	98.51										
90.10	99.68	88.01	100.73	86.04	101.70	84.18	102.59	82.45	103.38										
80.76	104.14	79.12	104.83	77.65	105.43	76.48	105.89	75.64	106.20										
114.46	100.83	107.91	103.32	103.63	105.47	100.31	107.25	97.66	108.70										
95.35	109.94	53.18	111.09	51.09	112.17	89.08	113.15	87.17	114.04										
85.25	114.87	83.37	115.63	81.65	116.27	80.26	116.73	79.27	117.05										
116.92	106.57	111.55	109.54	107.79	112.47	104.73	114.56	102.22	116.27										
99.98	117.78	97.82	119.18	95.70	120.53	93.62	121.79	91.60	122.96										
89.51	124.10	87.41	125.17	85.42	126.10	83.77	126.82	82.55	127.33										
119.58	110.32	114.59	114.43	111.62	117.42	108.82	119.88	106.48	121.89										
104.37	123.68	102.31	125.38	100.27	127.03	98.24	128.63	96.25	130.14										
94.19	131.66	92.07	133.17	90.05	134.54	88.35	135.65	87.07	136.45										
122.47	112.96	118.45	117.84	115.42	121.27	112.87	124.06	110.73	126.35										
108.78	128.40	106.88	130.37	104.99	132.29	103.12	134.17	101.29	135.99										
99.40	137.85	97.46	139.73	95.64	141.49	94.10	142.97	92.95	144.07										
125.79	115.15	122.18	120.72	119.47	124.54	117.18	127.64	115.25	130.19										
113.50	132.47	111.79	134.66	110.11	136.80	108.44	138.91	106.83	140.94										

105.19 143.03 103.54 145.14 102.02 147.12 100.76 148.76 99.90 149.99
129.61 117.20 126.40 123.33 124.01 127.51 122.00 130.89 120.31 133.66
118.78 136.14 117.29 128.51 115.83 140.84 114.35 143.11 113.01 145.30
111.60 147.53 110.20 149.76 108.93 151.82 107.90 153.50 107.15 154.73
133.66 119.01 130.67 125.56 128.81 130.03 127.09 133.65 125.65 136.60
124.34 139.24 123.08 141.77 121.83 144.24 120.60 146.65 119.42 148.96
118.22 151.30 117.02 153.63 115.91 155.77 115.00 157.51 114.33 158.78
137.67 120.48 135.31 127.35 133.59 132.05 132.15 135.84 130.94 138.94
129.85 141.70 128.80 144.34 127.76 146.93 126.74 149.45 125.75 151.87
124.74 154.32 123.72 156.77 122.76 159.02 121.97 160.86 121.37 162.21
141.61 121.62 139.70 128.75 138.30 133.60 149.02 132.78 151.62 140.76
135.28 143.61 134.43 146.34 133.60 149.02 132.78 151.62 132.00 154.13
131.19 156.68 130.38 159.24 125.62 161.61 128.98 163.56 128.50 164.99
145.56 122.49 144.09 129.83 143.02 134.83 142.11 138.86 141.36 142.15
140.65 145.07 140.04 147.86 139.41 150.59 138.80 153.25 138.23 155.80
137.55 158.41 137.09 161.02 136.58 163.46 136.17 165.49 135.88 167.00
149.73 123.14 148.71 130.63 147.97 135.75 147.33 139.86 146.81 143.20
146.33 146.17 145.88 148.95 145.46 151.74 145.05 154.41 144.68 156.97
144.33 159.56 144.01 162.14 143.75 164.54 143.57 166.49 143.47 167.93
154.28 123.57 153.81 131.16 153.42 136.37 153.07 140.55 152.77 143.94
152.49 146.94 152.23 149.77 151.99 152.52 151.76 155.17 151.57 157.68
151.41 160.22 151.28 162.72 151.19 165.03 151.13 166.89 151.11 168.24
158.74 123.65 158.86 131.33 158.83 136.63 158.74 140.87 158.64 144.30
158.53 147.32 158.42 150.17 158.32 152.91 158.22 155.53 158.14 158.00
158.07 160.45 158.05 162.83 158.10 164.95 158.26 166.63 158.48 167.84
164.32 123.13 165.27 131.03 165.68 136.47 165.90 140.81 166.02 144.30
166.11 147.37 166.16 150.25 166.15 153.02 166.21 155.68 166.20 158.18
166.15 160.67 166.05 163.11 165.86 165.34 165.55 167.17 165.17 168.56
167.88 122.42 165.40 130.55 170.07 136.11 170.47 140.52 170.73 144.07
170.92 147.17 171.07 150.09 171.20 152.90 171.30 155.59 171.38 158.14
171.44 160.70 171.48 163.24 171.52 165.59 171.59 167.53 171.67 168.93

171.98 121.27 174.02 129.80 174.94 135.52 175.52 140.02 175.91 143.63
176.22 146.74 176.48 149.74 176.70 152.56 176.91 155.30 177.09 157.86
177.26 160.43 177.44 162.96 177.61 165.27 177.75 167.13 177.86 168.47
176.23 119.93 178.58 128.90 179.70 134.77 180.43 139.37 180.94 143.04
181.35 146.23 181.70 149.22 182.01 152.09 182.29 154.82 182.55 157.39
192.81 159.95 183.07 162.46 183.32 164.73 183.53 166.57 183.68 167.89
180.69 118.60 183.11 127.85 184.41 133.90 185.27 138.59 185.89 142.32
186.37 145.57 186.79 148.60 187.17 151.49 187.51 154.23 187.82 156.80
185.18 117.42 187.68 126.73 189.14 132.90 190.13 137.69 190.84 141.50
188.13 159.35 188.44 161.83 188.72 164.06 188.97 165.84 189.16 167.10
191.40 144.80 191.83 147.87 192.30 150.80 192.68 153.57 193.01 156.15
193.33 158.70 193.64 161.16 193.93 163.36 194.16 165.10 194.34 166.32
189.57 116.22 192.32 125.54 193.96 131.78 195.09 136.67 195.83 140.56
196.52 143.93 197.04 147.07 197.50 150.06 197.89 152.87 198.23 155.49
198.55 158.07 198.83 160.54 199.07 162.74 199.26 164.48 199.41 165.71
193.76 114.82 196.99 124.14 198.87 130.50 200.15 135.52 201.04 139.52
201.73 142.99 202.30 146.21 202.78 149.28 203.19 152.17 203.52 154.85
203.81 157.48 204.05 160.01 204.25 162.26 204.39 164.04 204.49 165.31
197.80 113.07 201.65 122.56 203.81 129.09 205.26 134.26 206.24 138.39
207.00 141.98 207.62 145.31 208.12 148.49 208.53 151.48 208.85 154.25
209.11 156.98 209.31 159.55 209.45 161.92 209.53 163.76 209.58 165.08
201.77 111.02 206.30 120.80 208.75 127.53 210.37 132.89 211.48 137.18
212.31 140.90 212.97 144.38 213.51 147.69 213.93 150.82 214.24 153.72
214.47 156.57 214.62 159.30 214.70 161.73 214.71 163.65 214.68 165.03
205.65 108.82 210.83 118.90 213.58 125.88 215.39 131.44 216.62 135.91
217.53 139.80 218.26 143.44 218.83 146.90 219.27 150.18 219.58 153.24
219.79 156.24 219.97 159.13 219.92 161.71 219.87 163.73 219.78 165.16
209.42 106.54 215.15 116.95 218.20 124.19 220.20 129.97 221.55 134.62
222.56 138.68 223.35 142.45 223.98 146.12 224.44 149.58 224.77 152.80
224.97 155.99 225.05 159.06 225.03 161.80 224.95 163.97 224.87 165.48
213.09 104.06 219.37 114.92 222.71 122.44 224.90 128.45 226.39 133.29

227.49 137.53 228.37 141.51 229.06 145.33 229.57 148.96 229.93 152.37
 230.15 155.75 230.24 159.03 230.19 161.98 230.07 164.34 229.93 166.06
 216.51 101.61 223.42 112.87 227.05 120.67 229.43 126.91 231.06 131.94
 232.27 136.36 233.24 140.51 234.00 144.50 234.58 148.33 234.99 151.92
 235.26 155.50 235.38 158.99 235.36 162.16 235.22 164.72 235.00 166.56
 219.68 99.38 226.98 111.00 230.86 119.05 233.42 125.49 235.17 130.65
 236.49 135.26 237.54 149.57 238.39 143.73 239.04 147.71 239.51 151.46
 239.83 155.22 240.01 158.88 240.07 162.23 240.67 164.94 240.08 166.95
 223.05 56.72 230.94 108.85 235.08 117.19 237.83 123.85 239.73 129.24
 241.17 133.98 242.34 138.46 242.30 142.75 244.64 146.96 244.59 150.90
 244.98 154.84 245.21 158.69 245.31 162.15 245.36 164.92 245.36 166.95
 227.13 93.62 235.60 106.25 240.05 114.93 243.02 121.84 245.10 127.47
 246.70 132.36 248.02 137.03 249.12 141.56 250.01 145.95 250.68 150.12
 251.16 154.32 251.42 158.45 251.41 162.14 251.21 164.89 250.99 166.52
 230.92 50.55 239.84 103.82 244.56 112.61 247.72 119.94 249.96 125.71
 251.71 130.79 253.10 135.61 254.44 140.31 255.49 144.88 256.32 149.25
 255.97 153.72 257.39 158.22 257.47 162.48 257.14 165.17 256.40 169.18
 233.43 88.59 242.72 102.16 247.59 111.35 250.89 118.62 253.23 124.49
 255.09 129.67 256.67 134.59 258.04 139.38 259.21 144.06 260.18 148.55
 261.00 153.16 261.64 157.85 262.08 162.36 262.41 166.35 262.80 169.65
 236.98 85.70 246.55 99.63 251.61 109.28 255.06 116.83 257.55 122.83
 259.54 128.11 261.27 133.14 262.80 138.04 264.15 142.82 265.31 147.44
 266.37 152.18 267.31 156.58 268.13 161.50 268.85 165.20 269.46 167.80
 241.40 82.14 251.46 97.08 256.71 106.85 260.34 114.48 263.00 120.61
 265.17 126.01 267.09 131.13 268.83 136.13 270.41 141.02 271.82 145.72
 273.17 150.56 274.42 155.44 275.48 160.01 276.23 163.74 276.70 166.40
 246.09 78.52 256.25 94.32 261.63 104.37 265.41 112.16 268.22 118.35
 270.55 123.85 272.64 129.04 274.58 134.10 276.37 139.03 278.02 143.78
 279.64 148.65 281.21 153.56 282.64 158.23 283.75 162.27 284.51 165.59
 259.21 75.52 260.44 91.96 265.90 102.21 269.78 110.10 273.70 116.39
 275.16 121.89 277.39 127.11 279.45 132.18 281.47 137.12 283.31 141.87

285.14	146.73	286.64	151.59	288.66	156.06	290.31	159.64	292.01	162.13
257.99	70.54	287.66	88.01	273.15	98.51	277.16	106.52	280.25	112.86
282.89	118.37	285.34	123.57	287.69	128.61	289.65	132.91	292.10	138.20
294.26	143.02	286.37	147.86	288.15	152.45	289.41	156.39	299.90	159.65
263.43	67.55	272.47	85.48	277.93	96.07	282.01	104.11	285.18	110.44
287.91	115.52	280.48	121.08	292.97	126.06	295.40	130.87	297.75	135.48
300.16	140.22	272.59	145.02	304.86	149.55	306.85	153.45	308.59	156.28
270.87	64.61	276.99	82.14	284.42	92.75	288.54	100.79	291.78	107.08
294.61	112.50	287.29	117.55	299.93	122.37	302.55	127.00	305.13	131.40
307.86	135.89	310.72	140.41	313.45	144.75	315.80	148.51	317.39	151.55
276.70	62.60	284.52	79.32	289.95	89.91	294.09	97.93	297.35	104.17
300.21	109.50	302.64	114.42	305.64	119.08	308.35	123.50	311.05	127.63
313.96	131.77	317.09	135.84	320.31	139.62	323.20	142.77	325.82	145.14
283.15	59.55	291.62	75.67	297.09	86.23	301.20	94.23	304.44	100.40
307.27	105.61	309.97	110.36	312.65	114.79	315.33	118.90	318.01	122.64
320.88	126.25	323.95	129.61	327.02	132.44	329.77	134.47	331.92	135.65
293.63	52.72	303.93	69.11	309.46	79.97	313.40	88.05	316.41	94.16
318.99	99.22	321.42	102.73	323.80	107.83	326.14	111.53	328.40	114.79
330.74	117.82	333.10	120.56	335.29	122.81	337.02	124.45	338.23	125.54
311.18	39.23	322.18	60.03	327.08	72.18	330.23	80.77	332.52	87.04
334.41	92.09	336.14	96.48	337.79	100.28	339.28	103.84	340.88	106.84
342.44	109.61	343.92	112.10	345.31	114.16	346.44	115.69	347.26	116.69
•55874	•82934	•66146	•62751	•54720	•63700	•53982	•84178	•53315	•84602
•52608	•85043	•51818	•85527	•50931	•66058	•49959	•86626	•48919	•87218
•47777	•87848	•46548	•88506	•45242	•89181	•43656	•89968	•42036	•90736
•35693	•93413	•42444	•50546	•46227	•88674	•47007	•88263	•46893	•88323
•46395	•88605	•45545	•89026	•44507	•89550	•43298	•90140	•41569	•90767
•40495	•91434	•38916	•52116	•37303	•92782	•35552	•93467	•34041	•94028
•32546	•94556	•42488	•90525	•47296	•88108	•48564	•87416	•48606	•87393
•48043	•87704	•47064	•88232	•45732	•88930	•44094	•89754	•42198	•90665
•39946	•91675	•37418	•92736	•34870	•93724	•32838	•94455	•31865	•94787

•50507	•86308	•64598	•63780	•56252	•82651	•56566	•82464	•56195	•82717
•55454	•83216	•54412	•83901	•53086	•64746	•51495	•65722	•49652	•86802
•47475	•88012	•44595	•89305	•42469	•90534	•40419	•91468	•39439	•91894
•67306	•73959	•66458	•74722	•66204	•75105	•65526	•75541	•64908	•76072
•64190	•76579	•63357	•77368	•62404	•78139	•61340	•78977	•60173	•79870
•58854	•80847	•67390	•81892	•55660	•82944	•54340	•83947	•53351	•84579
•78695	•61702	•75866	•65145	•74237	•76999	•73400	•67915	•72754	•68607
•72184	•69206	•71641	•69768	•71116	•70302	•70621	•70800	•70174	•71244
•69771	•71638	•65438	•71961	•69198	•72191	•69056	•72327	•68987	•72393
•84300	•53792	•82353	•56728	•80907	•58771	•80049	•59534	•79487	•60678
•79071	•61219	•78745	•61638	•78501	•61948	•78352	•62137	•78316	•6218
•88964	•45666	•87453	•48498	•86305	•50512	•85614	•51675	•80368	•59506
•84873	•52883	•84655	•53231	•84510	•53461	•84439	•53573	•84444	•53565
•84532	•53426	•84711	•53142	•84560	•52742	•85216	•52328	•85375	•52068
•92952	•36877	•91428	•40508	•90487	•42568	•90000	•43590	•89684	•44235
•89461	•44684	•89294	•45018	•89164	•45274	•89064	•45471	•88984	•45627
•88911	•45770	•88833	•45921	•88725	•46128	•88524	•46513	•88284	•46968
•95285	•30345	•94232	•33472	•93618	•35151	•93292	•36009	•93079	•36554
•92930	•36933	•92814	•37224	•52716	•37466	•52626	•37690	•92529	•37927
•92407	•38223	•92244	•38614	•92032	•39117	•51764	•39741	•91520	•40301
•96857	•28873	•96325	•26862	•95948	•28179	•95735	•28894	•95605	•29319
•95524	•29582	•95472	•29748	•95440	•29852	•95418	•29922	•95356	•29992
•95361	•30105	•95296	•30310	•95193	•30631	•95068	•31018	•94950	•31254
•98010	•19852	•97870	•20525	•97660	•21506	•97524	•22116	•97452	•22431
•97422	•22561	•97424	•22553	•97453	•22424	•97507	•22188	•97582	•21858
•97678	•21425	•97793	•20896	•97920	•20289	•98061	•19595	•98182	•18981
•98908	•14740	•98992	•14156	•98894	•14832	•98803	•15426	•98759	•15702
•98751	•15756	•98773	•15620	•98823	•15296	•98902	•14781	•99005	•14072
•99135	•13123	•99287	•11923	•99446	•10514	•99603	•08897	•99715	•07544
•99708	•07632	•99744	•07145	•99681	•07983	•99617	•08743	•99582	•09136

.99570	.09268	.99578	.09175	.99607	.08860	.99655	.08295	.99721	.07464
.99800	.06318	.99881	.04881	.99947	.03243	.99991	.01310	.99556	.00911
.99970	.02429	.99995	.00535	.99992	.01282	.99970	.02448	.99950	.03150
.99936	.03569	.99928	.03788	.99927	.03826	.99933	.03665	.99947	.03258
.99971	.02411	.99997	.00743	.99974	.02281	.99757	.06968	.99486	.10128
.98997	.14129	.99558	.08955	.99836	.05725	.99932	.03938	.99961	.02796
.99981	.01972	.99992	.01295	.99998	.00666	1.00000	.00007	.99956	.00871
.99975	.02320	.99984	.04595	.99925	.08652	.98888	.14871	.98092	.19441
.97327	.22965	.98898	.14805	.99469	.10296	.99670	.08116	.99773	.06729
.99836	.05717	.99880	.04888	.99914	.04151	.99940	.03456	.99562	.02769
.99978	.02074	.99588	.01517	.99990	.01403	.99983	.01835	.99989	.01478
.995894	.28361	.98158	.19103	.99002	.14053	.99313	.11706	.99480	.10184
.99586	.09094	.99659	.08251	.99711	.07596	.99746	.07117	.99767	.06822
.99774	.06725	.99764	.06872	.99736	.07256	.99702	.07712	.99675	.08060
.99445	.29839	.97592	.21811	.98530	.17084	.98917	.14679	.99140	.13086
.99285	.11938	.99385	.11072	.99453	.10446	.99493	.10057	.99508	.09912
.99497	.10016	.99460	.10375	.99404	.10906	.99350	.11383	.99327	.11580
.96006	.27979	.97293	.23109	.98083	.19486	.98523	.17126	.98800	.15446
.98988	.14188	.99123	.13215	.99217	.12493	.99274	.12028	.99298	.11831
.99285	.11915	.99235	.12314	.99150	.13005	.99028	.13510	.98915	.14690
.96453	.26397	.99653	.24498	.97620	.21687	.98129	.19255	.98475	.17397
.98722	.15834	.99508	.14737	.99047	.13775	.99144	.13055	.99205	.12584
.99233	.12362	.99225	.12426	.99174	.12830	.99051	.13741	.98880	.14927
.96159	.27448	.96351	.26767	.97083	.23977	.97718	.21242	.98164	.19075
.98493	.17296	.98750	.15761	.98954	.14425	.99111	.13301	.99231	.12379
.99322	.11626	.99386	.11062	.99415	.10796	.99370	.11209	.99239	.12317
.94820	.31767	.99376	.30058	.96443	.26432	.99278	.23172	.97860	.20580
.98294	.18395	.98640	.16436	.98921	.14650	.99148	.13029	.99328	.11575
.99476	.10226	.99593	.09013	.99672	.08096	.99687	.07905	.99637	.08515
.92562	.37845	.94117	.33792	.95719	.28945	.96810	.25057	.97555	.21980
.98110	.19349	.99556	.16933	.98920	.14656	.99214	.12510	.99447	.10502

99634 -.08548 .99776 -.06693 .99871 -.05085 .99923 -.03917 .99945 -.03309
 89622 -.4436C .92700 -.37507 .94954 -.31363 .96321 -.26875 .97241 -.23327
 97926 -.20262 .58474 -.17401 .98922 -.14646 .99279 -.11986 .99555 -.09425
 99765 -.06851 .99908 -.04294 .99983 -.01821 .99997 .00771 .99953 .03056
 86685 -.49858 .91282 -.40835 .94200 -.33561 .95828 -.28595 .96917 -.24640
 97726 -.21206 .58372 -.17963 .98900 -.14754 .99315 -.11681 .99628 -.08622
 99850 -.05474 .99974 -.02265 .99996 .00905 .99912 .04150 .99769 .06791
 85598 -.51701 .90302 -.42959 .93490 .35491 .95341 .30167 .96579 .25931
 97500 -.22222 .98238 .18692 .98837 .15210 .99307 .11751 .99654 .08306
 99889 -.04705 .99995 -.00991 .99969 .02487 .99876 .04981 .99820 .06005
 85249 -.52275 .85583 -.44440 .92823 .37201 .94857 .31656 .96220 .27234
 97237 -.23345 .98055 .19626 .98721 .15940 .99245 .12262 .99631 .08582
 99889 -.04704 .99998 -.00628 .99943 .03375 .99776 .06684 .99628 .08622
 83039 -.55719 .88385 .46778 .92214 .38685 .94386 .33035 .95845 .28527
 96940 -.24551 .97827 .20735 .98554 .16942 .99130 .13160 .99557 .09404
 99848 -.05507 .99985 .01464 .99962 .02741 .99726 .07395 .99427 .10686
 82731 .56174 .87565 .47562 .91686 .35920 .93946 .34266 .95470 .29758
 96621 .25775 .97564 .21938 .98347 .18108 .98975 .14280 .99447 .10503
 99775 .06711 .99950 .03163 .99999 .00464 .99998 .00589 .99997 .00824
 81542 .57888 .87201 .48948 .91151 .41127 .93462 .35566 .95025 .31148
 96219 .27240 .97210 .23456 .98052 .19644 .98748 .15777 .99292 .11880
 99689 .07883 .99915 .24123 .99952 .01236 .99998 .00705 .99988 .01544
 80291 .59610 .96464 .50240 .90601 .42325 .92911 .36980 .94480 .32765
 55690 .29042 .96714 .25423 .97609 .21735 .98385 .17990 .99040 .13824
 99582 .09133 .99937 .03525 .99555 .02951 .99520 .09787 .98744 .15802
 80960 .58699 .86413 .50325 .90160 .43257 .92414 .38205 .93947 .34263
 95139 .30758 .96163 .27436 .97076 .24006 .97893 .20419 .98621 .16550
 99295 .11940 .99823 .05949 .99973 .02337 .99121 .13227 .97518 .20300
 79843 .60209 .95892 .51211 .99870 .43855 .92041 .39095 .93518 .35419
 94671 .32210 .95667 .25116 .96565 .25985 .97376 .22758 .98103 .19385
 98763 .15677 .99300 .11807 .99615 .08762 .99650 .08362 .99556 .08982

80272	59636	86002	51026	89583	44441	51604	40109	92978	36812
94054	31261	9498	18376	95832	28568	96596	25870	97273	23193
97866	20548	98297	18376	98438	17607	98173	19030	97651	21549
83179	59760	85830	51315	89319	44966	91094	41253	92300	38480
93245	36120	94065	33937	94806	31808	95475	29739	96070	27759
96614	25804	97116	23842	97599	21781	98034	19732	98343	18127
82272	56845	86571	50054	89152	45299	90628	42268	91639	40051
92410	38215	93085	36540	93693	34952	94240	33448	94720	32063
95137	30805	95479	29731	95829	28580	96486	26277	96849	24904
82930	55888	86722	49792	89070	45460	90213	43146	90987	41490
91579	40155	92082	36995	92531	37921	92940	36908	93304	35977
93597	35207	93632	35115	92833	37176	90009	43571	87539	4834
86681	49863	88012	47474	89045	45509	89594	44215	90120	43340
80410	42733	90620	42285	90787	41925	90947	41576	91151	41128
91498	40350	92196	39729	93617	35156	95943	28196	97113	23855
90070	43445	89046	45506	89036	45525	89325	44957	89472	44662
89493	44620	89418	44771	89274	45057	89106	45389	88980	45635
88981	45633	89168	45266	89291	45023	88477	44604	87805	47857
92147	38845	89657	44290	89003	45591	88974	45646	88829	45928
88531	46500	88077	47355	87469	48465	86729	45781	85900	51197
85034	52622	84349	45374	84323	53755	85462	51926	86235	50631
92608	37732	89746	44110	88526	45721	88767	46047	98411	46727
87842	47789	87033	49247	85951	51113	84566	53371	82847	56004
80644	59132	77884	62722	74684	66501	71381	70034	69442	71957
89096	45409	88837	45913	88977	45640	88870	45849	88386	46774
87594	48243	86466	50237	84939	52777	82942	55862	80363	59513
76860	63972	72071	69324	65803	75299	58212	81310	52378	85185
83846	54496	87963	47565	90023	43541	90256	43056	89834	43929
89011	45575	87807	47854	86183	50719	84105	54095	81540	57889
78316	62182	74425	66791	70267	71152	66849	74372	64624	76314
87400	48593	92811	37230	94366	33092	94735	32016	94562	32526

.94090 --.33868 .93360 --.35831 .92358 --.38340 .51067 --.41314 .89473 --.44661
.87471 --.48465 .85035 --.52622 .82288 --.56822 .79417 --.60770 .77273 --.63473

APPENDIX A

FORTRAN Listing of Program CONFORM

```

C PROGRAM CNFORM, DETERMINES TRANSFORMATION 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/FOR1A/ COB , COC , BZRO,XK,NMAX
C
C READ CONTINUATION FLAG, NUMBER OF SHORELINE COORDS, NUMBER OF
C SEAWARD BORY COORDS, NUMBER OF COEFFS, MAX NUMBER OF ITERATIONS,
C NUMBER OF INTEGRATION POINTS AND CONVERGENCE CRITERION
C READ(5,1)IWAIT, MQ, MQP, NMAX, JMAX1, IL, VAPWT
C 1 FORMAT (6F15, F5.3)
C
C
C
C READ COORDS DELINEATING SHORELINE
C DO 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 BDY
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

```

```

C
PI= 3.1415927
XLAMDA= X2(MQ)
XK= PI/XLAMDA

WRITE(6,9) MQ,MCP,XLAMDA,VARWT
FORMAT( 1H1 ,////
9//,20X,40NUMBER OF COORDS DELINEATING SHORELINE= ,I3
9//,20X , 4HNUMBER OF COORDS DELINEATING SEAWARD BDRY= ,I3
9 //,20X,32HORIZONTAL EXTENT OF SHORELINE= ,F7.2
9//,20X,29HCONVERGENCE VARIANCE, VARWT= ,F9.6)

WRITE(6,60) NMAX,JMAX,IJL
FCRMT( //,20X, 39HNUMBER OF TRANSFORMATION COEFFS ,NMAX= ,I3
9//,20X, 29HNUMBER OF ITERATIONS ,JMAXI= ,I3
9//,20X, 34HNUMBER OF INTEGRATION POINTS ,IJL= ,I3)

WRITE(6,70)
FORMAT( 1H1 , 62X,16HSHORELINE COORDS ,//// )
MQ13= MQ/2
MQ3= MC13*3
IADD= MQ-MQ3
DO 71 I=1,MQ13
J= I+MQ13
K= J+MQ13
71 WRITE(6,72) I,I,X2(I),Y2(I) , J,J,X2(J),Y2(J),K,K,X2(K),Y2(K)
72 FORMAT( /,3( 7X,3HX2(,I3,5H),Y2(,I3,3H)= ,F7.2,3X,F7.2) )
IF( IADD.EQ. 0 ) GO TO 75
IF( IADD.EQ. 1 ) NUMI= MQ
IF( IADD.EQ. 2 ) NUMI= MQ-1
DO 73 I=NUMI,MQ
73 WRITE(6,74) I,I,X2(I),Y2(I)
74 FORMAT( /, 85X,3HX2(,I3,5H),Y2(,I3,3H)= ,F7.2,3X,F7.2 )
75 CONTINUE

```



```

80 WRITE(6,80)
   FORMAT( 1H1          ,61X,19HSEAWARC BDRY CCOORDS  ,// )
   MQP13= MQP/3
   MCP3= 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(,I3,3F)= ,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(,I3,3F)= ,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 , BZFO
102 FORMAT( 2E14.7 )
DO 103 N=1,NMAX
103 READ(5,102) CCB(N) , CDC(N)
104 READ(5,104) WSX,WSY,WCX,WCY
105 FORMAT( 4E14.7 )
CONTINUE
C
A= -B
CALL COEFFS( XLAMDA , A , B , WSX,WSY,WCX,WCY )
C
C PUNCH 102, B,BZFO
C DO 107 N=1,NMAX
C107 PUNCH 102, CCB(N) , CDC(N)
C PUNCH 104, WSX,WSY,WCX,WCY
C
999 STOP
END
SUBROUTINE COEFFS( XLAMDA , A , B , WSX,WSY,WCX,WCY )
DIMENSION CCB(200) , CDC(200)
DIMENSION PLMB(200) , DUMC(200)
DIMENSION X2(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, MG
COMMON/YA/ X2P, Y2P, MQP
COMMON/JOHNR/ R
COMMON/JOHNS/ S
COMMON/JOHNT/ T
COMMON/LVM/ 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
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)= SORT( U(J) )
12  SUM= SUM + U(J)
  SUM= SUM -0.5*U(I) + 0.5*U(INC)
13  ARCYB(II)= ARCYB(I) + SUM*DELX
  ARCB= ARCYB(MQ)
  GO TO 19
15  CONTINUE
  ARCYB(I)= 0.0
  DO 18 I=1,MQ1
    II= I+1
    DELX= X2(II)-X2(I)
    DELX= DFLX*DELX
    DELY= Y2(II)-Y2(I)
    DELY= DELY*DELY
18  ARCYB(II)= ARCYB(I) + SORT( 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

```

```

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)= SGR1( 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
   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 )
29  ARCA= ARCYA(MGF)
C   CONTINUE
C
CALL CUR2YA( SLP1 , SLPN , SIGMA1 )
CALL CUR3YA( SLP1 , SLPN , SIGMA1 )
C
WRITE(6,30)
30  FORMAT( 1H1,/,45X,40HARLENGTH CF SEAWARD BDRY AND COASTLINE ,
9//,10X,25HARLENGTH OF SEAWARD BDRY,8X,3HX2P,9X,3HY2P
915X,22HARLENGTH CF COASTLINE,7X,2HX2,10X,2HY2 )
DO 31 IT=1,MQP
IK= IT
ALNA= ARCYA(IT)
ATXA(IT)= CUR4YA( ALNA , SIGMA1 , IK )
ATYA(IT)= CUR5YA( ALNA , SIGMA1 , IK )
DO 32 IT=1,MC
IK= IT
ALNB= ARCYE(IT)
ATXB(IT)= CLR4YB( ALNE , SIGMA , IK )
ATYB(IT)= CUREYB( ALNB , SIGMA , IK )
IF( MQ .GE. MQP ) NUM2= MQP
IF( MQ .LE. MQP ) 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. MQP ) GO TO 38
35  NUM1= MQP+1
DO 36 I=NUM1,MC

```

```

36 WRITE(6,37)
37 FORMAT( 84X,F8.2,7X,F7.2,5X,F7.2
        )
38 NUM1 = MC+1
DO 39 I=NUM1,MOP
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
Z(IT)= XIT*CELX I
45 NUM1= 1
NUM2= NMAX
DO 50 N=NUM1,NUM2
XN= N
XNK= XN*XK
DO 50 IT=1,IL1
XNKXI= XNK*Z(IT)
SN(N,IT)= SIN( XNKXI )
50 CS(N,IT)= COS( XNKXI )
C

```

```

55      ICOUNT= 1
      CONTINUE
      IF( ICOUNT .EQ. 1 ) GC TC 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*(IT)/XLAMDA
      ALNB= ARCB*(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= YSLMA + ATYA(IT)
      YSUMB= YSLMB + ATYB(IT)
      YSUMA= YSUMA - 0.5*ATYA(1) - 0.5*ATYA(IL)
      YSUMB= YSLWB - 0.5*ATYB(1) - 0.5*ATYB(IL)
      ADUM= YSUMA*FT*WSY
      BDUM= YSUMB*FT*WCY
      GO TO 81
C
76      CONTINUE
      A= #B
      ETAA= A
      ETAB= E
      SUMA= 0.0

```



```

SUMB= 0.0
DO 77 IT=1,IL
XI= Z(IT)
CALL SLFAC( XI,ETAA,DXDELTA,DYDELTA,SCFACA,SF)
CALL SLFAC( XI,ETAB,DXDELTA,DYDELTA,SCFACB,SF)
SUMA= SUMA + SCFACA
SUMB= SUMB + SCFACB
U(IT)= SCFACA
V(IT)= SCFACB
77 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= ARCE/ALNB
SUMA= 0.0
SUMB= 0.0
ALNA= 0.0
ALNB= 0.0
DO 78 IT=1,IL
IK= IT
ATXA(IT)= CURAYA( ALNA , SIGMAI , IK )
ATYA(IT)= CURYA( ALNA , SIGMAI , IK )
ATXB(IT)= CURAYB( ALNB , SIGMA , IK )
ATYB(IT)= CURBYB( 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

```

```

78      CONTINUE
C
75      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)
          YSUMB= YSLMB + ATYB(IT)
80      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 )/BOTTOM
        BERR= BTDLUM-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  
       IIT= IL-IIT+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  
       R(IT)= -ATXA(IT)*SINN  
       S(IT)=  ATXB(IT)*SINN  
       T(IT)=  ATYA(IT)*COSS  
       W(IT)=  ATYB(IT)*COSS  
       GO TO 86  
       SINN= SN(N,IT)  
       COSS= CS(N,IT)  
       R(IT)= ATXA(IT)*SINN  
       S(IT)= ATXB(IT)*SINN
```

```

T(IT) = ATYA(IT)*COSS
W(IT) = ATYB(IT)*COSS
CONTINUE
XASUMS = XASUMS + R(IT)
XBSUMS = XBSUMS + S(IT)
YASUMC = YASUMC + T(IT)
YBSUMC = YBSUMC + W(IT)
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 )
CHR = 'COSH( XNKB )
TB = SHB/CHB
SM = TB*TF
TM = SM/SHR
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

```

86

87

```

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

```

```

COC(N)= DUMC(N)
AMPB= AES( COB(N)*COSH(XNKB) )
AMPC= ABS( CFC(N)*SINH(XNKB) )
270 WRITE(6,271) N,COR(N),ERRB ,AMPB ,N,CCC(N),ERRC ,AMPC
271 FORMAT( 7X,2H(, I3,3F)= ,E14.7,3X,E10.3,3X,E10.3 = 27X
,92HC(, I3,3F)= ,E14.7,3X,E10.3,3X,E10.3 )
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 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,CYDETA,SCFACA,SF)
CALL SLFAC( XI,ETAB,DXDETA,DYDETA,SCFACB,SF)
SUMA= SUMA + SCFACA

```

```

SUMB= SUNE + 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
V(IT)= SCFACB
SUMEXA= SUMEXA + 0.5*SUMXA - 0.5*( ABS(ATXA(1)+Z(1)-R(1)) )**2
SUMEYA= SUMEYA - 0.5*SUMYF - 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
ERAR= EF/4.0
WSY= EFYA/EBAR
WCY= EFYE/EBAR
WSX= EFXA/EBAR
WCX= EFXB/ERAR
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= ARCE/ALNB

```

(Remove)
(Remove)

277

C

```

SUMA= 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)= ALNE
ATXA(IT)= CUR4YA( ALNA , SIGMA1 , IK )
ATYA(IT)= CUR5YA( ALNA , SIGMA1 , IK )
ATXB(IT)= CUR4YE( ALNE , 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 * XMCD A
ALNB= SUMB * XMCD B
278 CONTINUE
VARXA= VARXA / XIL
VARYA= VARYA / XIL

```

278

C


```

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,56H RATIO OF COASTLINE ARCLNGTH TO *TRANSFORM* GEN
GENERATED = ,F8.4
9/,32X,59H RATIO OF SEAWARD BDRY ARCLNGTH TO *TRANSFCRM* GENERATEC
9= ,F8.4 ,//, 66X,84H VARIANCE ,26X,7HER FNCT
9//,7X,48H TRANSFCRM COASTLINE FRM 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 FRM THAT SPECIFIED USING Y ,E14.7 ,
92X,E14.7,4X,E14.7
9/ ,71X,E14.7,20X,E14.7
WRITE(6,281) ←
291 FORMAT (//
,15X,42H SEAWARD BDRY COORDS , 0=XI=LAMDA , ETA=A,
,39H COASTLINE COORDS , 0=XI=LAMDA , ETA=B
923X
9/ ,9X,9HX(XI,ETA),2X,7FX=GIVEN,4X,9HY(XI,ETA),2X,7HY=GIVEN,4X
9RHARC=TRAN,4X,1HI
98X ,9HX(XI,ETA),2X,7FX=GIVEN,4X,8HY(XI,ETA),2X,7HY=GIVEN,4X
9RHARC=TRAN
)

C
C
DO 285 I=1,IL

```

```

285 WRITE(6,286) I,R(I),ATXA(I),S(I),ATYA(I),X(I)
91,T(I),ATXE(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
IF( ICOUNT.LE. JMAX1 ) GC TO 55
300 RETURN
375
END
FUNCTION XTRAN( XI , ETA )
DIMENSION COB(200) , COC(200)
COMMON/FOR1A/ COE , COC
XTRAN= 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)
XTRAN=XTRAN+(COC(N)*CSH +COC(N)*SNH )*SIN(XNKXI)
10 CONTINUE
XTRAN= XTRAN + XI
RETURN
END
FUNCTION YTRAN( XI , ETA )
DIMENSION COB(200) , COC(200)
COMMON/FOR1A/ COB , COC
BZRO,XK,NMAX

```

```

YTRAN= 0.0
XKA= X*ETA
XKXI= X*KXI
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)
CONTINUE
YTRAN= YTRAN + ETA + BZRO
RETURN
END
SUBROUTINE SLFAC( XI , ETA , DXDETA , DYDETA , SCFAC , SF )
DIMENSION CCE(200) , COC(200)
COMMON/FORIA/ CCB , COC
XKA= X*ETA
XKXI= X*KXI
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
+CC(N)*SNH
)*SIN(XNKXI)
YDETA= XN*XK*(COB(N)*SNH
+CC(N)*CSH
)*COS(XNKXI)
DXDETA= DXDETA + XDETA
DYDETA= DYDETA + YDETA

```

10

```

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
SLP1 = SLF1
SLPN = SLPN
1 SIGMAP = AES(SIGMA)*(N-1)/(X2(N)-X2(1))
DELS = SIGMAP*DELX1
EXPS = EXP(DELS)
SINHS = .5*(EXPS+1./EXPS)
SINHIN = 1./(DELX1*SINHS)
DIAG1 = SINHIN*(DELS*.5*(EXPS+1./EXPS)-SINHIN)
DIAGIN = 1./DIAG1
A(1) = DIAGIN*(DX1-SLP1)
SPDIAG = SINHIN*(SINHIN*DELS)
S(1) = DIAGIN*SPDIAG

```

```

IF (N.EQ.2) GC TO 3
DO 2 I = 2,NM1
  DELX2 = X2(I+1)-X2(I)
  DX2 = (Y2(I+1)-Y2(I))/DELX2
  DELS = SIGMAP*DELX2
  EXPS = EXP(DELX)
  SINHS = .5*(EXPS+1./EXPS)
  SINHN = 1./(DELX2*SINHS)
  DIAG2 = SINHN*(DELS*(.5*(EXPS+1./EXPS))-SINHS)
  DIAGN = 1./(DIAG1+DIAG2-SFCIAG* S(I-1))
  A(I) = DIAGN*(DX2-DX1-SPDIAG* A(I-1))
  SPDIAG = SINHN*(SINHS-DELS)
  S(I) = DIAGN*SPDIAG
  DX1 = DX2
2  DIAG1 = DIAG2
3  DIAGN = 1./(DIAG1-SPDIAG* S(NM1))
  A(N) = DIAGN*(SLPPN-DX2-SPDIAG* A(NM1))
DO 4 I = 2,N
  IBAK = NPI-I
  A(IBAK) = A(IBAK)+ S(IBAK)* A(IBAK+1)
  RETURN
5  IF (N.EQ.2) GC TO 6
  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 = -DELN/DELN/DELNM1
C3 = DELN/DELNN/DELNM1
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 , MC
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 CURIYB MUST BE CALLED EARLIER.
N= M0
S = X2(N)-X2(1)
SIGMAP = ARS(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(I) ) 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(SIGMAF*DEL1)
SINH1 = .5*(EXPS1-1./EXPS1)
EXPS = EXP(SIGMAF*DEL2)
SINH2 = .5*(EXPS-1./EXPS)
EXPS = EXPS1*EXPS
SINHS = .5*(EXPS-1./EXPS)
CURVYB = ( A(I)*SINH1 + A(I-1)*SINH2)/SINHS + ((Y2(I)-A(I))*DEL1 +
9(Y2(I-1) - A(I-1))*DEL2 )/DELS
I1 = I
RETURN
END
FUNCTION CURVYB( T , SIGMA , IT )
DIMENSION X2(150) , Y2(150)
DIMENSION A(150)
COMMON/JCHNA/ A
COMMON/YB/ X2, Y2 , MQ
C THIS FUNCTION DIFFERENTIATES THE COASTLINE AT A GIVEN VALUE
C FOR X. THE VALUE RETURNED IN CURVYB IS THE VALUE OF DY/DX AT X.
C SUBROUTINE CURIYE MUST BE CALLED EARLIER.
N = MQ
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(SIGMAP*DEL1)
COSH1 = .5*(EXPS1+1./EXPS1)
EXPS = EXP(SIGMAP*DEL2)
COSH2 = .5*(EXPS+1./EXPS)
EXPS1 = EXPS1*EXPS
SINHS = .5*(EXPS=1./EXPS)/SIGMAP
CURDYB= ( A(I)*COSH1- A(I=1)*COSH2)/SINHS+((Y2(I)-A(
1 I))-(Y2(I=1)-f(I=1)))/DELS
1 I I = 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/JOHNS/ 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)/(ARCYN)=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
C(I) = DIAGIN*(DX1-SLPP1)
SPDIAG = SINHIN*(SINHS-DELS)
S(I) = DIACIN*SPDIAG
IF (N.EQ.2) GO TO 3
DO 2 I = 2,NM1
  DELX2 = ARCYB(I+1)-ARCYE(I)
  DX2 = (X2(I+1)-X2(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))
  C(I) = DIACIN*(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 = NPI-I
4  C(IEAK) = 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 = ARCYB(NM1)-ARCYE(N-2)
DELNN = ARCYB(N)-ARCYB(N-2)
C1 = (DELNN+DELN)/DELNN/DELN
C2 = -DELNN/DELN/DELNM1
C3 = DELN/DELNN/DELNM1
SLPEN = 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/JOHNC/ C
  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 CUR4YB IS THE VALUE OF X
C AT ITS PARTICULAR ARCLENGTH. SUBROUTINE CUR2YE MUST BE CALLED BEFORE.
N= MG
S = ARCYE(N)-ARCYE(1)
SIGMAP = ABS(SIGMA)*(N-1)/S

```

```

IF (IT.EG.1) I1 = 2
1 DO 2 I = I1,N
  .IF( ARCYE(I)-T ) 2,2,3
2 CONTINUE
  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 = ARCYR(I)-ARCYB(I-1)
  EXPS1 = EXP(SIGMAF*DEL1)
  SINHD1 = .5*(EXPS1-1./EXPS1)
  EXPS = EXP(SIGMAF*DEL2)
  SINHD2 = .5*(EXPS-1./EXPS)
  EXPS = EXPS1*EXPS
  SINHS = .5*(EXPS-1./EXPS)
  CUR4YR= ( C(I)*SINHDI+ C(I-1)*SINHDI2)/SINHS+((X2(I)-C(I))*DELI +
9 (X2(I-1) - C(I-1))*DEL2 )/DELS
  I1 = I
  RETURN
END
SUBROUTINE CUR3YR( SLP1 , SLPN , SIGMA )
  DIMENSION ARCYB(150)
  DIMENSION X2(150) , Y2(150)
  DIMENSION C(150) , S(408)
  COMMON/JOHNS/ 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
SLPPI = 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) = CIAGIN*(DX1-SLPPI)
SPDIAG = SINHIN*(SINHS-DELS)
S(1) = DIAGIN*SPDIAG
IF (N.EG.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)-SFDIAG* 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*(SLPFN=DX2-SPDIAG* D(NM1))
DO 4 I = 2,N
   IBAK = NPI=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 O(150)

```

```

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 CURSYB IS THE VALUE OF Y
C AT ITS PARTICULAR ARCLENGTH. SUEROUTINE CUR3YE MUST BE CALLED BEFORE.
N= MQ
S = ARCYE(N)-ARCYB(1)
SIGMAP = ABS(SIGMA)*(N-1)/S
IF (IT.EO.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(I) ) 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)
  CURSYB= ( D(I1)*SINHD1+ D(I-1)*SINHD2)/SINHS+(Y2(I)-D(I))*DELI +
9 (Y2(I-1) - C(I-1))*DEL2 )/DELS
I1 = I
RETURN
END

```

```

SUBROUTINE CURIYA( 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 X2F
  N= MQP
  NM1 = N-1
  NP1 = N+1
  DELX1 = X2P(2)-X2P(1)
  DX1 = (Y2F(2)-Y2F(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
  EXPS = EXP(DELS)
  SINHS = .5*(EXPS+1./EXPS)
  SINHN = 1./(DELX1*SINHS)
  DIAG1 = SINHN*(DELS*.5*(EXPS+1./EXPS)-SINHS)
  DIAGIN = 1./DIAG1
  R(1) = DIAGIN*(DX1-SLPP1)
  SPDIAG = SINHN*(SINHS-DELS)
  S(1) = DIAGIN*SPDIAG
  IF (N.EC.2) GO TO 3
  DO 2 I = 2, NM1
    DELX2 = X2P(I+1)-X2P(I)
    DX2 = (Y2F(I+1)-Y2F(I))/DELX2
    DELS = SIGMAP*DELX2
    EXPS = EXP(DELS)

```

```

SINH5 = .5*(EXPS=1./EXPS)
SINHIN = 1./((DELX2*SINH5)
DIAG2 = SINHIN*(DELS*(.5*(EXPS+1./EXPS))=SINH5)
DIAGIN = 1./((DIAG1+DIAG2=SPDIAG* S(I=1))
B(I) = DIAGIN*(DX2=DX1=SPCIA* B(I=1))
SPDIAG = SINHIN*(SINH5=DELS)
S(I) = DIAGIN*SPDIAG
DX1 = DX2
2— DIAG1 = DIAG2
3 DIAGIN = 1./((CIAG1=SFCDIAG* S(NM1))
B(N) = DIAGIN*(SLPPN=DX2=SPDIAG* B(NM1))
DO 4 I = 2,N
IBAK = NFI=I
4 B(IRAK) = B(IBAK) = S(IFAK)* B(IBAK+1)
RETURN
5 IF (N.FG.2) 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/DELNN/DELNM1
C3 = DELN/DELNN/DELNM1
SLPPN = C3*Y2P(N=2) + C2*Y2P(NM1)+ C1*Y2P(N)
GO TO 1
6 B(I) = 0.

```



```

B(2) = 0.
RETURN
END
FUNCTION CURVYA( T = SIGMA , IT )
DIMENSION X2P(150) , Y2P(150)
DIMENSION B(150)
COMMON/JOHNP/ R
COMMON/YA/ X2F , Y2F , 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)
SIGMAP= ABS(SIGMA)*(N-1)/S
IF (IT.EG.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(I) ) 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(SIGMA*DEL2)
  SINHD2 = .5*(EXPS+1./EXPS)
  EXPS = EXPS1*EXPS
  SINHS = .5*(EXPS+1./EXPS)

```

```

CURVYA= ( E(I)*SINHDI + B(I-1)*SINH2) / 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)
COSHDI = .5 * (EXPS1 + 1. / EXPS1)
EXPS = EXP(SIGMAP * DEL2)
COSH2 = .5 * (EXPS + 1. / EXPS)

```

```

EXPS = EXPS1*EXPS
SINHS = .5*(EXPS+1./EXPS)/SIGMAP
CURDYA = ( B(I)*COSH(1- B(I-1))*COSH(D2)/SINH(S+((Y2P(I)-B(
1) )-(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/YA/ ARCYA
COMMON/YA/ X2P , Y2P , MGF
C FITS SPLINE--X2P AS A FUNCTION OF ARCYA
N = MGF
NM1 = N-1
NP1 = N+1
DELX1 = ARCYA(2)-ARCYA(1)
DX1 = (X2P(2)-X2P(1))/DELX1
IF (SIGMA.LT.C.) GO TO 5
SLP1 = SLP1
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(I) = DIAGIN*(DX1-SLPP1)
SPDIAG = SINHIN*(SINH5=DELS)
S(1) = DIAGIN*SPDIAG
IF (N.EG.2) GC TO 3
DO 2 I = 2,NM1
  DELX2 = ARCYA(I+1)=ARCYA(I)
  DX2 = (X2P(I+1)-X2P(I))/DELX2
  DELS = SIGMAP*DELX2
  EXPS = EXP(CELS)
  SINHS = .5*(FXFS=1./EXPS)
  SINHIN = 1./(DELX2*SINHS)
  DIAG2 = SINHIN*(DELS*(.5*(EXPS+1./EXPS)+1./EXPS))=SINH5
  DIAGIN = 1./(DIAG1+DIAG2-SFCIAG* S(I-1))
  E(I) = DIAGIN*(DX2-DX1=SPDIAG* E(I-1))
  SPDIAG = SINHIN*(SINH5=CELS)
  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,N
  IBAK = NPI=I
  E(IBAK) = E(IBAK) = S(IBAK)* E(IBAK+1)
RETURN
5  IF (N.EG.2) GC 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 = C2*X2P(N-2) + C2*X2P(NM1) + C1*X2P(N)
GO TO 1
6 E(1) = 0.
  F(2) = 0.
  RETURN
FND
FUNCTION CURAYA( T , SIGMA , IT )
  DIMENSION ARCYA(150)
  DIMENSION X2P(150) , Y2P(150)
  DIMENSION E(150)
  COMMON/JCHNE/ E
  COMMON/AYA/ X2P , Y2P , MQP
C THIS FUNCTION INTERPLATES THE SEAWARD PCRY AT A GIVEN VALUE FOR
C THE ARCLENGTH. THE VALUE RETURNED IN CUR4YA IS THE VALUE CF 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)
SINHDI = .5*(EXFSI-1./EXPS1)
EXPS = -EXP(SIGMAP*DEL2)
SINHDI2 = .5*(EXPS-1./EXPS)
EXPS = EXPS1*EXPS
SINHS = .5*(FXPS-1./FXPS)
CUR4YA = ( E(I)*SINHDI+ E(I-1)*SINHDI2)/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/YA/ ARCYA
COMMON/YA/ X2P , Y2P , 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

```

```

SLPP1 = SLP1
SLPPN = SLPN
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 = NPI-I

```

```

4   F(IRBK) = F(IEAK) - S(IEAK)* F(IEAK+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*Y2P(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/DELMN1
   C3 = DELN/DELNN/DELMN1
   SLPPN = C3*Y2P(N-2)+ C2*Y2P(NM1)+ C1*Y2P(N)
   GO TO 1
6   F(1) = 0.
   F(2) = 0.
   RETURN
END
FUNCTION CUR5YA( T , SIGMA , IT )
DIMENSION ARCYA(150)
DIMENSION X2P(150) , Y2P(150)
DIMENSION F(150)
COMMON/JCHNF/ F
COMMON/YA/ X2P , Y2P , MQP

```

C THIS FUNCTION INTERPLATES THE SEAWARD BDRY AT A GIVEN VALUE FOR
C THE ARCLNGTH. THE VALUE RETURNED IN CUR5YA IS THE VALUE OF Y
C AT ITS PARTICULAR ARCLNGTH. SUEROUTINE CUR3YA MUST BE CALLED BEFORE.


```

N= MOP
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 .OP. 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
C DIMENSION COB(200) , COC(200)
C DIMENSION X2(1000) , Y2(1000)
C DIMENSION X2P(1000) , Y2P(1000)
C DIMENSION ATXA(1000) , ATYA(1000) , ATXB(100) , ATYB(1000)
C DIMENSION XI(47) , ETA(15) , X(47,15) , Y(47,15) , Z(47,15)
C DIMENSION SX(100) , SY(100)
C DIMENSION A(5) , B(5) , C(5) , IIC(5)
C COMMON /YE/ X2,Y2,MQ
C COMMON /YA/ X2P,Y2P,MQP
C COMMON/XIETXY/ XI,ETA,X,Y
C COMMON/SXY/ SX,SY
C COMMON/ED/ XKBETA,BZPBT,BZMBT
C COMMON /FORIA/ COB,COC,BZPC,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
C READ(5,1)NMAX,NUMXI,NUMETA,DELSS,SSTRT,DELT,ND,NS
C I FORMAT(3I4, 3F5.1, 2I4)
C
C SLMAP=60./51.
C XLAMDA= 360.

```

XIDUM= 360.0*26.0/33.0

PI = 3.141593

XK = PI/XLAMDA

SIGMA = .1.

G= 32.2

NUMET1= NUMETA=1

NUMXI1= NUMXI=1

NUMXI2= NUMXI=2

C

READ (5,27) BETA,BZRO

DC 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 REPRESENTIVE OF COASTLINE.

C*** ARCLENGTH.

C*** T* AXIS IS PARALLEL TO ETA AND IS REPRESENTIVE OF LONG WAVE

C*** TRAVEL TIME.

C

C

C*** DETERMINE TRANSFORM GENERATED COASTLINE AND SEA BDRY

C

N3=1

N4= 121

DELXI= XLAMDA/(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
  CONTINUE
85  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 ARCLNGTH OF CCASTLINE AND SEA BDRY
C*** AS A FUNCTION OF (EVENLY SPACED) XI.
C
Y2(1) = 0.
X2(1) = 0.
X2P(1) = C.
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= AES( ATXA(II)-ATXA(I) )
DELY= ABS( ATYA(II)-ATYA(I) )
X2P(II)= SLMAP*SORT( DELX*DELX + DELY*DELY )
X2P(II)= X2P(II) + X2P(I)
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( IHI, //, 46X, 35#REGION---LAGUN MADRE TC MARSH ISLAND )
39 FORMAT ( //, 55X, 21#COASTLINE COORDINATES, // )
40 FORMAT ( 62X, 8#CRIGINAL, // // )
49 FORMAT ( //, 51X, 28#SEAWARD BOUNDARY COORDINATES )
50 FORMAT ( 53X, 24H( 100 FM DEPTH CONTOUR ) , // )
59 FORMAT ( 54X, 23#INTERPCLATION BY SPLINE, // // )
69 FORMAT( // )
79 FORMAT( 56X, 15#TRANSFCRM GENERATED , // // )
89 FORMAT( 5CX, 30#ARCLENGTH, XI ALONG COASTLINE, // // )
99 FORMAT( 47X, 37#ARCLENGTH, XI ALONG SEAWARD BOUNDARY, // // )

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

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

```

```

113 WRITE(6,113)
    FORMAT ( //,53X,33HSTRETCHED SHELF COORDINATE SYSTEM
    ,
    9/,39X,52HEVENLY SPACED IN S-STAR (DISTANCEF PARALLEL TC 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 GPID.
C USER SUPPLIES EXPANSION CCEFFS 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)= 0.6058735E+03
E(1)=0.5479735E+05
C(1)=0.9044300E+00
A(2)=0.1355315E+03
E(2)=1.5
C(2)=1.
A(3)= 0.478846E+03
E(3)=0.3038714E+07
C(3)=0.1707654E+01
A(4)=0.
E(4)=1.
C(4)=1.
A(5)= 0.2870103E+03
E(5)= 0.4707750E+11
C(5)= 0.5160152E+01

```



```

WRITE(6,114)
114  FORMAT(20X,11HD S PL/D S* ,5X,6HS=STAR,7X,10HS=PARALLEL,11X,?HXI,
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 ,11,3X,6HA,B,C= ,3(2X,F14.7),//)
K2= K2+IIC(I)
K3= K2+1
DO 119 J=K1,K3
K=J
XDUM= SSSTRT + (J-1)*DELSS
CALL CURVS( XDUM,SDUM,DDUM,A1,B1,C1 )
XI(J)= CURV2( SDUM,SIGMA,K )
WRITE(6,117) DDUM,XDUM,SDUM,XI(J),J
117  FORMAT( 30X,F8.4,10X,F6.1,5X,F6.1,5X,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 ISLINE OF XI ( XIDUM ).

```

```

C
DELETA= 2.0*BETA/(NS=1)
X2(I)= 0.
WRITE(6,25)
WRITE(6,79)
WRITE(6,999) XIDUM
FORMAT( 41X,33HARCLENGTH (N MI),ETA  ALCNG XI=  ,F8.3,/// )
999 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 ) GC TC 94
JJ= J+1
DELX= ABS( X2F(J)-X2F(JJ) )
DELY= ABS( Y2P(J)-Y2P(JJ) )
X2(JJ)= X2(JJ) + SLMAP*SQRT( DELX*DELX + DELY*DELY )
CONTINUE
94 DO 98 I=1,NS
WRITE(6,97) I,X2(I),Y2(I) ,X2P(I),Y2P(I)
FORMAT( 42X,I2,2X,F6.1,1H,,F7.3,10X,F6.1,1F,,F6.1 )
97 CONTINUE
98
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.
NDI= ND-1
DELS= X2(NS)/NDI

```

```

Y2P(I)= 0.
DO 123 I=1,NC
  READ(5,124) SY(I)
  J=I+1
  IF( I .EQ. NC ) GC TO 123
  Y2P(J)= Y2P(I) + DELS
123 CONTINUE
124 FORMAT( F8.2 )
C DETERMINE LONG WAVE TRAVEL TIME (MIN) ALCNG XI=XIDUM.
  X2P(I)= 0.0
  DO 128 I=1,ND1
    J=I+1
    F1= 1.0/SCFT( G*SY(I))*6.0 )
    F2= 1.0/SCRT( G*SY(J))*6.0 )
    X2P(J)= X2P(I) + (F1+F2)*(Y2P(J)-Y2P(I))*5C.0
128 CONTINUE
    TIME= X2P(NC)
C
C MGP= NC
  CALL CURV3( SLP1,SLPN,SIGMA I
C
  WRITE(6,29)
  WRITE(6,131) XIDUM
131 FORMAT( 7CX,3HXI=,F6.1/,13X,8HD T/D T*,12X,6HT=STAR .6X,4HTIME .
    96X,8HS=NORMAL ,8X,3HETA ,10X, SHDEPTH, // )
  DO 133 I=1,NC
133 WRITE(6,124) X2P(I),Y2P(I),SY(I)
134 FORMAT( 43X,F6.1,6X,F6.1,21X,F6.1 I
C
C TOTAL OF IIC(IRG)=NUM CF INCRFMENTS OF DEL-T*.
C EXPANSION CURVE= T= A+E*(T*)**C

```

C

```
IRG= 2
IIC(1)= 4
IIC(2)= 10
TIM1= IIC(1)*DELTT
TIM2= IIC(2)*DELTT
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)*DELTT
CALL CURV9( WDUM,XDUM,DDUM,A1,B1,C1 )
```

```

YDUM= CURV4( XDUM,SIGMA, K )
ZDUM= CLR2( YDUM,SIGMA,K )
ETA(J)= ZDUM
IF( J .EQ. 1 ) FTA(J)=BETA
IF( J .EQ. NUMETA ) ETA(J)= BETA
WRITE(6,143) DDUM,WDUM,XDUM,YDUM,ZDUM, J
143 FFORMAT( 11X,F8.4,12X,F6.1,6X,F6.1,7X,F7.3,2X,I2 )
145 CCNTINUE
KI= 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 FCINT 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 SHFCR( DELSS,NUMXI, DELTT,NUMETA, SLMAP )
C
C PUNCH X,Y COORDINATES HERE AS READ IN PROGRAM SURGE 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
      WRITE(6,200)
      FORMAT( 1H1, //, 50X, 35PHURRICANE CARLA COMPUTING GRID DATA )
      WRITE(6,201)
      FCRMAT( //, 60X, 15HSCALE FACTOR MU , // )
      DO 204 I=2, NUMXI1
      J=I-1
      SX(J)= ( XI(I+1)-XI(I-1) )/2./DELSS
      WRITE(6,202) J, SX(J)
      FORMAT( 60X, 3FMU( ., I2, 3H)= , E14.7 )
      204 CCNTINUE
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
      WRITE(6,200)
      WRITE(6,210)
      FORMAT( //, 60X, 15HSCALE FACTOR NU , // )
      SY(1)= (ETA(2)-ETA(1))/DELTT
      SY(NUMETA)= (ETA(NUMETA)-ETA(NUMET1))/DELTT
      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./DFLTT
      211 WRITE(6,212) J, SY(J)
      212 FCRMAT( 60X, 3FNU( ., 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
C*** COMPUTE SCALE FACTOR F AS READ IN PROGRAM SSURGE WITH NAME S.
C RECALL F IS S CRT CF AREA IN X,Y PLANE / MU*NU*DELS*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
C IQUAD= 4
C JQUAD= 2
C KCP= 1
C KOP= 2
C CALL TRAN2( KCP, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0. )
C
C WRITE(6,200)
C WRITE(6,300)
C FORMAT(/,60X,14HSCALE FACTOR F ,// )
C 300 DC 330 I=1,NUMXII
C DELXI= (XI(I+1)-XI(I))/IQUAD
C FRST= XI(I) +DELXI/2.
C DO 310 K=1,JQUAD
C X2(K)= FRST + (K-1)*DELXI
C 310 DC 330 J=1,NUMETI
C DELETA= (ETA(J+1)-ETA(J))/JQUAD
C FRST= ETA(J) + DELETA/2.
C SUM= 0.
C DO 320 K=1,IQUAD
C DO 320 L=1,JQUAD
C DUMETA= FRST + (L-1)*DELETA
C CALL TRAN2( KOP,X2(K),DUMETA,0.,DF1,DF2 )

```

```

SUM= SUM + DF1*CF1 + DF2*CF2
CONTINUE
320 X(I,J)= SUM*DELXI*DELETA
CONTINUE
330 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+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( 5(I4,2X,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,NUMXI=2 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,DF1,DF2 )
Z(I,J)= ATAN2( DF2,DF1 )
CCNTINUE
N= NUMETA
DO 410 I=2,NUMXII
K= I-1
X(K,I)= COS((Z(I=1,I)+2.*Z(I,2)+Z(I+1,1))+ 4.*Z(I,1))/8.)
Y(K,I)= SIN((Z(I=1,I)+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,NUMETI
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
DC 440 J=1,NUMETA
WRITE(6,3F1) J
WRITE(6,4F30) ( I,X(I,J),Y(I,J),I=1,NUMXI2)
430 FORMAT( 5(I3,2X,F8.5,1X,F8.5,4X) )
440 CCNTINUE
C
C PUNCH COS AND SIN THETA HERE AS READ IN FRCGFW SSURGE WITH
C NAMES COSG AND SING. THESE VALUES ARE IN ARRAYS X AND Y FOR
C I=1,2==NUMXI=2 AND J=1,2==NUMETA.
C
STOP
END
SUBROUTINE XUT( X,Y,N1,N2 )

```

```

DIMENSION X(1000) , Y(1000)
M= N2-N1+1
M4= M/4
N4= M4*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. 3 ) 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( 10CX,I4,2X,F6.1,1X,1F,,1X,F6.1,6X )
25 CONTINUE
RETURN
END .
SUBROUTINE SHRCR( DFLXI,NUMXI, DELETE,NUMETA,SLMAP )
DIMENSION XI(47) , ETA(15) , X(47,15) , Y(47,15)
DIMENSION SX(100) , SY(100)
COMMON/IEITX/ XI,ETA,X,Y
COMMON/SXY/ SX,SY
NUMPG= NUMXI/5
IADD= NUMXI-E*NUMPG
NPAGE= 1
DO 40 I=1,NUMXI
DO 40 J=1,NUMETA

```

```

CALL TRANI( XDUM, YDUM, XI(I), ETA(J) )
X(I,J)= XDUM
Y(I,J)= YDUM
CONTINUE
CONTINUE

40      WRITE(6,900) CELXI , DELETE
55

N1= NPAGE*5 -4
N2= NPAGE*5
IF( N2 .GT. NUMXI ) GC TC 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
DC 70 K=1, NUMETA
J= NUMETA-K+1
JJ= J-1
DC 60 I=I1,N2
IF( I .EQ. NUMXI ) GO TO 62
I= I+1
DELX= ABS( X(I1,J)-X(I,J) )
DELY= ABS( Y(I1,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,J), X(I2,J), Y(I2,J), SX(I2,J),
9X(I3,J), Y(I3,J), SX(I3,J), X(I4,J), Y(I4,J), SX(I4,J), X(I5,J), Y(I5,J),
9SX(I5,J) , J

```

```

64 GC TO 66
   CONTINUE
   WRITE(6,905) ETA(J),X(I1,J),Y(I1,J),SX(I1),X(I2,J),Y(I2,J),SX(I2),
66 9X I3,J),Y(I3,J),SX(I3),X(I4,J),Y(I4,J),SX(I4),X(I5,J),Y(I5,J),J
   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)= SLMAP*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 CCNTINUE
    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= ABS( X(I1,JJ)-X(I1,J) )

```

```

DELY= ABS( Y(I,JJ)-Y(I,J) )
SY(I)= SLMAP*SQRT( DELX*DELY + DELY*DELY )
WRITE(6,912) SY(I1)
CCONTINUE
GO TO 150
102 CONTINUE
110 CONTINUE
I1= N1
I2= N1+1
WRITE(6,914) XI(I1),XI(I2),I1,I2
DC 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*DELY + 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*DELY + DELY*DELY )
CONTINUE
112 CONTINUE
WRITE(6,916) SY(I1),SY(I2)
CCONTINUE
114 GO TO 150
120 CONTINUE
I1= N1
I2= N1+1
I3= N1+2
WRITE(6,917)XI(I1),XI(I2),XI(I3),I1,I2,I3
CC 128 K=1,NUMETA

```

```

J= NUMETA=K+1
JJ= J-1
DO 122 I=11,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 12E
DO 126 I=11,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 CONTINUE
WRITE(6,919) SY(I1),SY(I2),SY(I3)
GO TO 150
128 CONTINUE
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=11,I4

```



```

914  FORMAT( 2X,2(14X,3HXI=F7.2),/ ,2(19X,1F(,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,1F(,I2,1H),1X),// )
918  FORMAT( 1X,4PETA=F7.3,2X,F6.1,1H,,F6.1 , )
919  92(3X,F4.1,4X,F6.1,1F,,F6.1), / .5X,1F(,I2,1H) )
920  FORMAT( 3(18X,F4.1,2X),/ )
921  FORMAT( 2X,4(14X,3HXI=F7.2),/ ,4(15X,1H(,I2,1H),1X),// )
922  FORMAT( 1X,4PETA=F7.3,2X,F6.1,1H,,F6.1 , )
923  93(3X,F4.1,4X,F6.1,1H,,F6.1), / .5X,1H(,I2,1H) )
      RETURN
      END
      SUBROUTINE TRAN( XCST,YCST,XSEA,YSEA,XI )
      DIMENSION COB(200), CCC(200)
      COMMON/ED/ XKBETA,BZPBT,BZMBT
      COMMON/FCRIA/ CCB,CCC,EZRC,XK,NMAX
      XCST= 0.
      YCST= 0.
      XSEA= 0.
      YSEA= 0.
      XKXI= XK*XI
      DO 10 N=1,NMAX
      XN=N
      XNKBT= XN*XKEETA
      XNKXI= XN*XKXI
      CSHE= COSH( XNKBT )
      SNAH= SINH( XNKBT )
      A= COC(N)*CSH
      B= COB(N)*SNH
      C= CCC(N)*SNH

```



```

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

```

10

```

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

```

```

END
SUBROUTINE CURV9( XDCUM,Y,Y1 ,A,B,C )
Z= ABS( XDCUM )
W= ABS( C-1. )
Y1= B
Y= A+B*(XDCUM)**C
IF( Z .LT. 0.0001 .OR. W .LT. 0.001 ) GO TO 10
Y1= B*C*(XDCUM)**(C-1.)
CONTINUE
RETURN
END
SUBROUTINE CURV1( SLP1 , SLPN , SIGMA )
DIMENSION X2(1000) , Y2(1000)
DIMENSION R(800) , S(800)
COMMON/YF/ X2 , Y2 , MG
COMMON/JOHN1/ R
COMMON/JOHN2/ S
C *****
C THIS SUBROUTINE DETERMINES THE PARAMETERS NECESSARY TO
C COMPUTE AN INTERPLATARY 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 N= MC IS THE NUMBER OF VALUES TO BE INTERPOLATED ( .GE. 2 )
C X2 IS AN ARRAY CF N INCREASING ABSCISSAE OF THE FUNCTIONAL VALUES
C Y2 IS AN ARRAY CF 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 CF LENGTH AT LEAST N. ( SCRATCH STORAGE ).
C
C SPLN AND SLPN CONTAIN THE DESIRFD VALUES FOR THE FIRST
C DERIVATIVE CF THE CURVE AT X2(1) AND X2(N), RESPECTIVELY.
C IF THE QUANTITY SIGMA IS NEGATIVE THESE VALUES WILL BE
C DETERMINEC INTERNALLY AND THE USER NEED ONLY FURNISH
C PLACE-HOLDING PARAMETERS FOR SLP1 AND SLPN. SUCH PLACE-
C HOLDING PARAMETERS WILL BE ICMCRD BUT NOT DESTROYED.
C
C SIGMA CGNTAINS 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 SPLINF. 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 INFLT CF NCT. IF SIGMA IS NEGATIVE THE ENDDOINT
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 PROPCRTIONAL TO THE SECCND DERIVATIVE
C OF THE CURVE AT THE GIVEN NODES ( X2,Y2 ) .
C

```

```

C   N (MG), X2, Y2, SLF1, SLFN, SIGMA ARE UNALTERED.
C
C
C
C
C   N = MG
C   NM1 = N-1
C   NP1 = N+1
C   DELX1 = X2(2)-X2(1)
C   DX1 = (Y2(2)-Y2(1))/DELX1
C
C   DETERMINE SLOPES IF NECESSARY
C
C   IF (SIGMA.LT.0.) GO TO 5
C   SLPP1 = SLP1
C   SLPPN = SLPN
C
C   DENORMALIZE TENSION FACTOR
C
C   I 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
C   DELS = SIGMAP*DELX1
C   EXPS = EXP(DELS)
C   SINHS = .5*(EXPS+1./EXPS)
C   SINHN = 1./(DFLX1*SINHS)
C   CIAG1 = SINHN*(DELS*.5*(EXPS+1./EXPS))-SINHS)
C   DIAGIN = 1./DIAGI
C   R(1) = CIAGIN*(DX1-SLPP1)

```

```

C      SPDIAG = SINHN*(SINHS=DELS)
      S(I) = CIAGIN*SPCIAG
C
      IF (N.EQ.2) GO TO 3
C
      DC 2 I = 2,NM1
      DELX2 = X2(I+1)-X2(I)
      DX2 = (Y2(I+1)-Y2(I))/DELX2
      DELS = SIGMAP*DELX2
      EXPS = EXP(DELS)
      SINHS = .5*(EXPS+1./EXPS)
      SINHN = 1./DELX2*SINHS
      DIAG2 = SINHN*(DELS*(.5*(EXPS+1./EXPS))=SINHS)
      R(I) = DIAGIN*(DX2=DX1=SPDIAG* S(I=1))
      SPDIAG = SINHN*(SINHS=DELS)
      S(I) = CIAGIN*SPCIAG
      DX1 = DX2
      2  DIAG1 = CIAG2
C
      3  DIAGIN = 1./DIAG1=SPCIAG* S(NM1))
      R(N) = DIAGIN*(SLPPN=DX2=SPDIAG* R(NM1))
C
C  PERFORM BACK SUBSTITUTION
C
      DC 4 I = 2,N
      IBAK = NFI-I
      4  R(IBAK) = R(IBAK)- S(IBAK)* R(IBAK+1)
C
      RETURN
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 ENDPPOINTS.
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)
      GC 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

```

```

DIMENSION X2(1000) , Y2(1000)
DIMENSION R(800)
COMMON/YB/ X2, Y2 , WC
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 ON INPUT==
C
C T (X=COORD) CONTAINS A REAL VALUE TO BE MAPPED INTO THE CURVE
C NF (MQ) IS THE NUMBER OF POINTS WHICH WERE INTERPOLATED TO
C DETERMINE THE CURVE.
C X2,Y2 ARE ARRAYS CONTAINING THE AECISSAS AND ORDINATES OF THE
C ORIGINAL INTERPOLATED POINTS.
C R IS AN ARRAY WITH VALUES PROPORTIONAL TO THE SECOND
C DERIVATIVE OF THE CURVE AT THE NODES (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 PREVIOUSLY (WITH N,X2,
C Y2,R, AND SIGMA UNALTERED) AND THAT THIS VALUE OF T
C EXCEEDS THE PREVIOUS 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 SEQUENCE

```



```

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 ON OUTPUT==
C
C CURV2 (=Y=COORD) CONTAINS THE INTERPOLATED VALUE.
C FOR T .LT. X2(1), CURV2= Y2(1). FOR T .GT. X2(N),CURV2= Y2(N).
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 SEARCH FOR INTERVAL
C
C 1 DO 2 I = I1,N
C IF( X2(I)=T ) 2,2,3
C 2 CONTINUE

```

```

C
C      I = N
C CHECK TC INSURE CORRECT INTERVAL
C
C      3 IF( X2(I-1) .LE. T .OR. T .LE. X2(1) ) GO TO 4
C
C RESTART SEARCH AND RESET I1
C ( INPUT 'IT' WAS INCORRECT )
C
C      I1 = 2
C      GO TO 1
C
C SET UP AND PERFORM INTERPOLATION
C
C      4 DEL1 = T-X2(I-1)
C      DEL2 = X2(I)-T
C      DELS = X2(I)-X2(I-1)
C      EXPS1 = EXP(SIGMAP*DEL1)
C      SINHD1 = .5*(EXPS1-1./EXPS1)
C      EXPS = EXP(SIGMAP*DEL2)
C      SINHD2 = .5*(EXPS-1./EXPS)
C      EXPS = EXPS1*EXPS
C      SINHS = .5*(EXPS-1./EXPS)
C      CURV2 = ( R(I)*SINHD1+ R(I-1)*SINHD2)/SINHS+((Y2(I)-R(I))*DEL1
C      9(Y2(I-1) - R(I-1))*CEL2 )/DELS
C
C      I1 = I
C      RETURN
C      END
C SUBROUTINE CURV3( SLP1 , SLPN , SIGMA )
C DIMENSION X2F(1000) , Y2P(1000)

```

```

DIMENSION V(800), S(800)
COMMON/JCHN2/ S
COMMON/JOHN4/ V
COMMON /Y/ 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 *****
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 ABSCISSAE 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 PROPORTIONAL TO THE SECOND DERIVATIVE
C OF THE CURVE AT THE GIVEN NCOES ( X2P,Y2P )
C
C N(MQP), X2P,Y2P,SLP1, SLPN, SIGMA ARE UNALTERED.
C

```

```

N= MQP
NM1 = N-1
NP1 = N+1
DELX1 = X2P(2)=X2P(1)
DX1 = (Y2F(2)-Y2P(1))/DELX1
IF (SIGMA.LT.0.) GO TC 5
SLPP1 = SLPI
SLPFN = SLFN
1 SIGMAP = ABS(SIGMA)*(N-1)/(X2F(N)-X2P(1))
DELS = SIGMAP*DELX1
EXPS = EXP(DELS)
SINHS = .5*(EXPS+1./EXPS)
SINHIN = 1./(DELX1*SINHS)
DIAG1 = SININ*(DELS*.5*(EXPS+1./EXPS)-SINHS)
DIAGIN = 1./DIAG1
V(1) = DIAGIN*(DX1-SLPP1)
SPDIAG = SINHIN*(SINHS-DELS)
S(1) = DIAGIN*SPDIAG
IF (N.EC.2) GC TC 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(I-1))
  V(I) = DIAGIN*(DX2-DX1-SPDIAG* V(I-1))
  SPDIAG = SINHIN*(SINHS-DELS)
  S(I) = DIAGIN*SPDIAG

```

```

DX1 = DX2
2  DIAG1 = CIAC2
3  DIAGIN = 1./(CIAG1=SPDIAG* S(NM1))
   V(N) = DIAGIN*(SLPPN=DX2=SPDIAG* V(NM1))
DC 4  I = 2.*N
      IBAK = NPI-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 = X2F(N)=X2P(NM1)
   DELNM1 = X2F(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*Y2F(N=2) + C2*Y2P(NM1) + C1*Y2P(N)
   GO TO 1
6  V(1) = 0.
   V(2) = C.
   RETURN
END
FUNCTION CURVA( T , SIGMA , IT )
DIMENSION X2P(1000) , Y2P(1000)
DIMENSION V(EQC)

```

C

```

C
C *****
C COMMON/JCHN4/ V
C 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 *****
C ON INPUT==
C T {X=COORD} 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 ABSCISSAS AND ORDINATES OF THE
C ORIGINAL INTERPLATED 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 ON OUTPUT==
C CURV4 {Y=COORD} 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 .CR. T .LE. X2P(1) ) GC TC 4
  I1 = 2
GC TO 1
4  CEL1 = T-X2P(I-1)
  CEL2 = X2P(I) - T
  DELS = X2P(I)-X2P(I-1)
  EXPS1 = EXP(SIGMAP*DEL1)
  SINHD1 = .5*(EXFS1-1./EXFS1)
  EXPS = EXP(SIGMAP*DEL2)
  SINHD2 = .5*(EXPS-1./EXPS)
  EXPS = EXPS1*FYPS
  SINHS = .5*(EXPS-1./EXPS)
  CURV4 = ( V(I)*SINHC1+ V(I-1)*SINH2)/SINHS+((Y2P(I)-V(I))*DEL1 +
9(Y2P(I-1) -V(I-1))*CEL2 )/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,C1X1,C1ETA,DX1,DETA,DELT,F,IM,JM, IMM,JMM, N
COMMON /BLK3/ CCRIO,CS,C3,C4,ROT,IMM2,IMN3,JMM2
COMMON/BLK3/ GRAV, NMAX,IH1,JH1,IT1,IT1,IH2,
1JH2,IT2,IT2,IH3,JF3,IT3,JT3,IT4,JH4,IT4,JT4,INC
COMMON/BLK6/IH5,JH5,IT5,JT5,IH6,JH6,IT6,JT6
CCOMMON /HURRI/ YRANGE ,THIT ,XHIT ,YHIT
3 C10 ,C11 ,PHI ,
4 NT1 ,NT2 , PINF ,IX ,IY
5 DATA IM,JM, NMAX,INC/
45,15, 1321,10/
1 DATA DX I, DETA, DELT, GRAV, F/
1 13081.3,17441.7, 180.0, 9.8,.0025/
1 18, 15, 18, 14, 10, 15, 10, 14/
1 DATA IH3,JH3,IT3,JT3,IH4,JH4,IT4,JT4/
1 16, 15, 16, 14, 32, 15, 32, 14/
1 DATA IH5, JH5, IT5, JT5, IH6, JH6, IT6, JT6/
1 36, 15, 36, 14, 38, 15, 38, 14/

```

```

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

```

C

```
C4 = C3*C3
RGT = 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 ZFRO
CALL FIELD
DO 100 N=1 ,NMAX
IF((N/2)*2.E0.N)GC 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),
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,
11JH2,IT2,IT3,IH3,JH3,IT3,JT3,IH4,JH4,IT4,JT4,INC
COMMON/BLK5/DSDXI(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6
11(300),VCM5(300),VCM6(300)
COMMON/BLK7/HORS1(67),HORS2(67),HORS3(67),HORS4(67),HORS5(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),ROT(50),RAC(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.
CCSG(I,J)=0.
SING(I,J)=0.

```

```

100  CCNTINUE
      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

```

```

HOB51(K) = 0.
HOB52(K) = 0.
HOB53(K) = 0.
HOB54(K) = 0.
HOB55(K) = 0.
HOB56(K) = 0.
CONTINUE
500 DO 600 K=1,50.
TIM(K) = 0.
RCT(K) = 0.
RAD(K) = 0.
VRMAX(K) = 0.
PZRO(K) = 0.
CONTINUE
600 DO 700 K=1,150
TIMPOS(K) = 0.
XPOS(K) = 0.
YPOS(K) = 0.
CONTINUE
700 RETURN
END
SUBROUTINE FIELD
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,DELTA,F,IM,JM, IMM,JMM, N
COMMON/BLK5/DSDXI(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6
1(300),VCM5(300),VCM6(300)
COMMON/PLK7/HOB51(67),HOB52(67),HOB53(67),HOB54(67),HOB55(67),HOB5
16(67)
COMMON /TRANS/ XX(45,15) ,YY(45,15) ,CCSG(45,15),SING(45,15)

```

```

COMMON /HURF/  TIM(50) ,ROT(50) ,RAD(50) ,VRMAX(50)
1  PZRO(50) ,TIMPOS(150),XPOS(150),YPOS(150)
COMMON /HURR1/ YRANGE ,THIT ,XHIT ,YHIT ,
3  C10 ,C11 ,PHI ,
4  NT1 ,NT2 , PINF ,IX ,IY
5
C
READ(5,11) NT1
FORMAT( I5 )
11 READ(5,12)(TIM(I),RCT(I),RAD(I),VRMAX(I),PZRO(I),I=1,NT1)
12 FORMAT( SF10.1)
READ(5,11) NT2
13 READ(5,13)(TIMPOS(I),XPOS(I),YPOS(I),I=1,NT2)
FORMAT(3F10.1)
WRITE(6,15)
15 FORMAT(1H1,15X,24HTHE STORM PARAMETERS ARE ,//,1X,
15H TIME ROTATION RADIUS MAX WINDS CEN PRESS ,/,1X,
250H (HRS) (DEG) (NM) (KNOTS) (MB) ,/)
WRITE(6,16)(TIM(I),ROT(I),RAD(I),VRMAX(I),PZRO(I),I=1,NT1)
16 FORMAT(1X,F8.0,F10.0,2F10.1,F10.0)
WRITE(6,17)
17 FORMAT(1H1,15X,37HPE POSITIONS FOR THE STORM TRACK ARE ,//,1X,
129H TIME X COORD Y COORD ,/,1X,
210H (HRS) ,/)
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

```

```

DC 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 FGRMAT(E14.7)
600 CONTINUE
DC 700 J=1,JM
READ(NU1,8) DTDFI(J)
8 FORMAT(2X,E14.7)
700 CONTINUE
READ(NU1,9)(HCRS1(II),II=1,67)
READ(NU1,9)(HOBSS2(II),II=1,67)
READ(NU1,9)(HCBS3(II),II=1,67)
READ(NU1,9)(HCBS4(II),II=1,67)
READ(NU1,9)(HOBSS(II),II=1,67)
READ(NU1,9)(HCBS6(II),II=1,67)
9 FORMAT(19F4.1)
DC 300 II=1,67
HCRS1(II)= 59999.
HOBSS1(II)=HOBSS1(II)*0.3048
HOBSS2(II)=HOBSS2(II)*0.3048
HOBSS3(II)=HOBSS3(II)*0.3048
HCBS4(II)=HOBSS4(II)*0.3048
HOBSS5(II)=HCBS5(II)*0.3048
HOBSS6(II)=HOBSS6(II)*0.3048
300 CCNTINUE
DC 101 I=1,IM
READ (5,10) (XX(I,J),YY(I,J),J=1,JM)
10 FORMAT (10F7.2)

```

```

101 CCNTINUE
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.E0.IJ)GO TO 110
   GRID2(I,J)=GRID2(I,J)*0.3048*6.0
   GO TO 100
110 CCNTINUE
   GRID1(I,J)=0.
   GRID2(I,J)=C.C
100 CCNTINUE
200 CCNTINUF
CCCC*****
   WRITE(6,210)
210 FORMAT(1H1,////,20X,16HD E P T H S (M) ,///,7X,5HS E A ,58X,9HC 0
      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 FCRMAT(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.E0. IM ) GO TO 215
   WRITE(6,214)K,(GRID2(K,J),J=1,JM,2)
214 FORMAT(1X,2HI=,I2,8(4X,F6.1),/ )

```



```

CONTINUE
RETURN
END
SUBROUTINE ELEV
COMMON/BLK1/GRID1(45,15), GRID2(45,15),
1(3),HYD2(300),HYD3(300),HYD4(300),UCM1(300),UCM2(300),UCM3(300),UCM
2(45,15),HYD1(30
2(45,15),HYD2(300),HYD3(300),HYD4(300),UCM1(300),UCM2(300),UCM3(300),UCM4(300)
COMMON/FLK2/C2,CC,C1X,I,C1ETA,DX1,DELTA,DELTA,F,IM,JM, IMM,JMM, N
COMMON /BLK2C/ CCRIC,CS,C3,C4,RCI,IMM2,IMM3,JMM2
COMMON/BLK5/DSPXI(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6
1(300),VCM5(300),VCM6(300)
COMMON/STCRM/KC,XEYF,VEYF,P(45,15),WIND(46,15)

DC 180 J=3,JMM
DTET=DTDEI(J)*DETA
IS= 4
IMS= IMM3
IF((J/2)*2.E0.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=DSOXI(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 )

```

C

```

130 CONTINUE
180 CONTINUE
CC***
      DETE= DTDET(JM)*DETA
      DO 170 II=2,IMM,2
      IPI=II+1
      IMI= II-1
      C9= DELT/S(II,JM)/S(II,JM)
      DSXI= .DSDXI(II)*DXI
      URIT= GRID1(IPI,JM)*S(IPI,JM)
      ULEF= GRID1(IMI,JM)*S(IMI,JM)
      IF( II .EQ. 2 .OR. II .EQ. IMM ) ULEF=URIT
      VBOT= GRID2(II,JMM)*S(II,JMM)
      VTOP= -VBOT
      GRID1(II,JM)=CFIDI(II,JM)=C9*((URIT-ULEF)/DSXI+(VTOP-VBOT)/DETE)
170 CONTINUE
      C
      DC 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***
      DC 150 II=2,IMM
      IF((II/2)*2.EQ.II) GO TO 140
      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)= BARC
150 CONTINUE
RETURN
END
SUBROUTINE FLUX
COMMON/ELK1/GRID1(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/ELK2/C2,CC,C1X1,C1ETA,DX1,DETA,DELTA,F,IM,JM, IMM,JMM, N
COMMON /BLK20/ CCR10,CS,C3,C4,PC1,IMM2,IMM3,JMM2
COMMON/BLK5/DSDXI(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6
1(300),VCM5(300),VCM6(300)
COMMON/STCRM/NO,XEYE,YEYE,P(45,15),WIND(46,15)
C
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 = IMM2
IF((J/2)*2.EQ.J) IS=4
IF(IS.EG.4) IMS=IMM3
JP1 = J+1
JM1 = J-1
CC 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)/DSDXI(I)
CETA= C1ETA/S(I,J)/DDET(J)
D1 = GRID1(IM1,J)=GRID2(IM1,J)
D2 = GRID1(IP1,J)=GRID2(IP1,J)
D3 = GRID1(IM1,J)=GRID2(IM1,J)
D4 = GRID1(IP1,J)=GRID2(IP1,J)
DELHX = GRID1(IP1,J)=GRID1(IM1,J)=(P(IP1,J)-P(IM1,J))
DELHY = GRID1(IP1,J)=GRID1(IM1,J)=(P(IP1,J)-P(IM1,J))
Q = SORT(GRID1(I,J)*GRID1(I,J)+GRID2(I,J)*GRID2(I,J))
AVHX = C.5*(D1+D2)
AVHY = C.5*(D3+D4)
DBAR = C.5*(AVHY+AVFX)
G = (1.C+C2*F*Q)/(DBAR*DBAR)).
G2 = G*G
A = GRID1(I,J)+RCT*GRID2(I,J)=CXI*AVHX*DELFX+C2*WIND(IWX,J)
B = GRID2(I,J)=ROT*GRID1(I,J)=CETA*AVHY*DELHY+C2*WIND(IWY,J)

```

```

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=C1X1/(S(I,JM)*DSDX1(I))
DELHX = GRIC1(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 .EG. J ) GO TO 8100
GO TO 8500
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*(GRIC1(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( II .EQ. IMM ) III= IMM2
UBAR= 0.5*( GRID1(II,J+1)+GRID1(III,J-1) )
UFLUX= UBAR + GRIC1(II,J)

```

```

E = GRID2(II,J)-ROT*UFLUX=CET*AVHY*DELHY+C2*WIND(II,J)
GRID2(II,J)=B/G
GRID1(II,J)=UBAR
GC TO 9000
8900 CONTINUE
III= 1
III= 3
IF( II .EQ. IMM ) III= IM
IF( II .EQ. IMM ) III= IMM2
GRID1(III,J)= GRID1(III,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
G1= SORT(GRID1(II,1)**2+GRID2(II,1)**2)
G2= SORT(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/(O1**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 = CXI/S(II,1)/DSDXI(II)
CETA = CETA/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,1)-C3*GRID1(I,2)-CETA*D2*DELHY+C2*WIND(I,2)
BP = BB+.5*C3*GRID1(II,3)

```

```

DENOM = G1*G2+.25*C4
GRID1(I1,1) = (AP*G2+BP*.5*C3)/DENOM
GRID2(I,2) = (BP*G1-AP*.5*C3)/DENOM
GRID1(I,2) = .5*(GRID1(I,1)+GRID1(I1,1))
GRID2(I1,1) = .5*(GRID2(I,2)+GRID2(I1+IS,2))
CONTINUE
9050 GRID1(1,1) = GRID1(3,1)
      GRID1(IM,1) = GRID1(IMM2,1)
      RETURN
      END
SUBROUTINE DRAW1
COMMON/BLK1/GRID1(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,C1X1,C1ETA,DXI,DETA,DELT,F,IM,JM, IMM,JMM, N
COMMON /BLK20/ CORIO,CS,C3,C4,ROT,IMM2,IMM3,JMM2
COMMON/BLK3/ GRAV,
1JH2,IT2,JT2,IH3,JH3,IT3,JT3,IH4,JH4,IT4,JT4,INC
COMMON/PLK5/DSCXI(45),DTDET(15),HYDS(300),HYDE(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
NF = 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 METEF

C
IF (MOD(MIN,60) .NE. 0) RETURN
C CALCULATE THE X AND Y VELOCITIES
C AT THE INTERIOR POINTS
DO 100 J=2,JMW
IS = 3
IMS = IMM2
IF ((J/2)*2 .EQ. J) IS = 4
IF (IS .EQ. 4) IMS = IMN3
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
IM = 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
DC 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) GC TC 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= .5*(D1+D2)
UVEL(II,J) = GRID1(II,J)/DBAR
VVEL(II,J) = GRID2(II,J)/DBAR

```

```

305 GO TO 310
CONTINUE
JP1 = J+1
JM1 = J-1
D3 = GRID1(I,JP1)-GRID2(I,JP1)
D4 = GRID1(I,JM1)-GRID2(I,JM1)
CBAR = 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 CCNTINUE

T = MIN/6C.
WRITE (6,400) T
400 FCORMAT (1H1,6X,13FTIME (FRS) = ,F5.2,///,20X,12HE L E V (M),
1///,7X,5HS E A,58X,9HC O A S T,/)
WRITE (6,405) (J,J=2,JMM,2)
405 FCORMAT (8X,7(6X,2HJ=,I2),/ )
WRITE (6,406) (J,J=1,JM,2)
406 FCORMAT (4X,3(6X,2HJ=,I2),/)
DO 415 I=1,IM,2
K = I+1
WRITE (6,410) I,(GRID1(I,J),J=2,JMM,2)
410 FCORMAT (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)

```

C

```

411 FORMAT (1X,2I=,I2,8(4X,F6.2),/ )
415 CONTINUE
WRITE (6,419) T
419 FORMAT (1H1,6X,I3FTIME (PRS) = ,F6.2,///,20X,35HX AND Y. V E L O C
I I T I F S (CM/S),///,7X,5HS E A,58X,9HC C A S T,/)
WRITE (6,415) (J,J=2,JMM,2)
WRITE (6,406) (J,J=1,JM,2)
DO 425 I=1,IM,2
K = I+1
WRITE (6,421) I,(LVEL(I,J),J=1,JM,2)
WRITE (6,422) (VVEL(I,J),J=1,JM,2)
421 FORMAT (1X,2I=,I2,8(4X,F6.1))
422 FORMAT (5X,8(4X,F6.1),/ )
IF (I.EG. IM) GO TO 425
WRITE (6,423) K,(LVEL(K,J),J=2,JMM,2)
WRITE (6,424) (VVEL(K,J),J=2,JMM,2)
423 FORMAT (1X,2I=,I2,3X,7(4X,F6.1))
424 FORMAT (8X,7(4X,F6.1),/ )
425 CCNTINUE
RETURN
END -
SUBROUTINE METER
COMMON/BLK1/GRID1(45,15),GRID2(45,15), S(45,15),HYD1(30
10),HYD2(30),HYD3(30),HYD4(30),LCM1(300),UCM2(300),UCM3(300),UCM
24(300),VCM1(300),VCM2(300),VCM3(300),VCM4(300)
COMMON/BLK2/CP,CC,C1X1,C1ETA,CX1,DETA,DELT,F,IM,JM, IMM,JMM, N
COMMON/BLK3/ GRAV, NMAX,IH1,JH1,IT1,JT1,IH2,
1JH2,IT2,IT3,IH3,IT3,JT3,IF4,JH4,IT4,JT4,INC
COMMON/BLK5/DSDXI(45),DTDET(15),HYD5(300),HYD6(300),UCM5(300),UCM6
1(300),VCM5(300),VCM6(300)
COMMON/ELK6/IF5,JF5,IT5,JT5,IF6,JF6,IT6,JT6

```

C

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

```

120 UCM2(NUV) = GRIC1(IT2,JT2)/AVEDX
    VCM2(NUV) = GRID2(IT2,JT2)/AVEDY
    GO TO 10
130 UCM3(NUV) = GRIC1(IT3,JT3)/AVEDX
    VCM3(NUV) = GRID2(IT3,JT3)/AVEDY
    GO TO 10
140 UCM4(NUV) = GRIC1(IT4,JT4)/AVEDX
    VCM4(NUV) = GRID2(IT4,JT4)/AVEDY
    GO TO 10
150 UCM5(NUV)=GRIC1(IT5,JT5)/AVEDX
    VCM5(NUV)=GRID2(IT5,JT5)/AVEDY
    GO TO 10
160 UCM6(NUV)=GRIC1(IT6,JT6)/AVEDX
    VCM6(NUV)=GRID2(IT6,JT6)/AVEDY
10 CCNT INUE
    RETURN
    END
SUBROUTINE HUV
COMMON/BLK1/GRID1(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,DXI,DELTA,DELTA,F,IM,JM,
COMMON/BLK3/
    GRAV,
    NMAX,IH1,JH1,IT1,IT1,IH2,
1 JH2,IT2,JT2,IF3,JF3,IT3,JT3,IH4,JH4,IT4,JT4,INC
COMMON/BLK5/DSCXI(45),DTDET(15),HYD5(300),HYDA(300),UCM5(300),UCM6
1(300),VCM5(300),VCM6(300)
COMMON/BLK6/IH5,JH5,IT5,JT5,IF6,JH6,IT6,JT6
COMMON/BLK7/HOBS1(67),HOBS2(67),HOBS3(67),HOBS4(67),HCBSS(67),HOB
16(67)

```

```

WRITE (6,505) IH1,JH1,IH2,JH2,IH3,JH3,IH4,JH4,IH5,JH5,IH6,JH6

```

C

```

WRITE (6,506) IT1,JT1,IT2,JT2,IT3,JT3,IT4,JT4,IT5,JT5,IT6,JT6
505 FCRMAT(11-1,41+ ADJUSTED=COMPUTED HYDROGRAPH (M) AT ,6(I2,1H,,
6 I2,7X),/,30H AND THE OBSERVED WATER LEVEL ,// )
506 FCRMAT(11X,34+SIMULATED 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)*DELTA/3600.
MIN= (K-1)*DELTA/60. +1.CE=05
C CORRECT COMPUTED H FOR INITIAL OBSERVED WATER LEVEL CONDITIONS
HYD1(I)= HYC1(I)+1.
HYD1(I)= HYD1(I)+HOB51(I)
HYD2(I)= HYD2(I)+HOB52(I)
HYD3(I)= HYD3(I)+HOB53(I)
HYD4(I)= HYD4(I)+HOB54(I)
HYD5(I)= HYD5(I)+HOB55(I)
HYD6(I)= HYD6(I)+HOB56(I)
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)HOB51(J),HOB52(J),HOB53(J),HOB54(
9J),HOB55(J),HOB56(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)=,F6.2,5F + ,6(F6.2,6X))
511 FCRMAT (39X,4FV ,6(F7.2,5X))
512 FCRMAT (39X,4FV ,6(F7.2,5X))
513 FCRMAT( 34X,7HOB5 H ,6(F6.2,6X))

```

```

RETURN
END
SUBROUTINE WINDF
COMMON/BLK2/C2,CC,C1XI,C1ETA,DXI,DETA,DELTA,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) ,RAC(50) ,VRMAX(50) ,
1 PZRO(50) ,TIMPOS(150),XPOS(150),YPOS(150)
COMMON /HURR1/ YRANCE ,THIT ,XHIT ,YHIT ,
3 C10 ,C11 ,PHI ,
4 NT1 ,NT2 ,
5 COMMON /EXTRA/ X(45,15) ,Y(45,15) PINF ,IX ,IY
COMMON/ STCRM /NO,XEYE,YEYE,P(45,15),WIND(46,15)
C
DATA IT1P1,IT2P1/2,2/
C
NC=N
TIME=FLCAT(N-1)*DELTA/3600.
DC 101 I=IT1P1,NT1
IF(TIM(I),GE,TIME) GO TO 102
101 CONTINUE
I=NT1
IT1=I-1
102 DO 103 I=IT2P1,NT2
IF(TIMPCS(I),GE,TIME)GC TC 104
103 CONTINUE
I=NT2
IT2= I-1
104 IT1P1=IT1+1
IT2P1=IT2+1
TFAC1=(TIME-TIM(IT1))/(TIM(IT1P1)-TIM(IT1))

```

```
TFAC2={TIME-TIMPCS(IT2)}/(TIMPOS(IT2P1)-TIMPOS(IT2))
DEL T2=(TIMPCS(IT2F1)-TIMPCS(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))
PQ=PZRC(IT1)+TFAC1*(PZRO(IT1P1)-PZRO(IT1))
XEYE=XPOS(IT2)+TFAC2*(XPCS(IT2P1)-XPOS(IT2))
YEYE=YPCS(IT2)+TFAC2*(YPOS(IT2P1)-YPOS(IT2))
US=(XPOS(IT2P1)-XPOS(IT2))/DEL T2
VS=(YPOS(IT2P1)-YPOS(IT2))/DEL T2
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
CC 107 J=1,IY
XP=XX(I,J)
YP=YY(I,J)
XP1=XP=XEYE
YP1=YP=YEYE
RSMU=SGRT(XP1**2 + YP1**2)
IF(MOD(I+J,2).EQ.1)GO TO 109
A=YP1*CI0 -XP1*CI1
```



```

B= XPI*CI0 -YPI*CI1
IF(RSML.LT.1.0F=10) RSML= 1.0F=10
IF(RSML.LE. R)GC TO 105
C
C POINT IS OUTSIDE RADIUS TO MAX WINDS
C
S1=R/(R+RSML)
S2=(VR/RSML)*(R/RSML)**.45
GC TO 106
C
C POINT IS INSIDE RADIUS TO MAX WINDS
C
C
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
GC TO 107
109 RDI=22.*.84167
PP=P0+(FINF=P0)*EXP(-RCIST/RSML)
P(I,J)=(FINF=PP)*.01
107 CONTINUE
108 CCNTINUE

```

```

DO 110 J=1, IY
DC 110 I=1, IX, 2
IF((J/2)*2.E0.J)GC TO 112
WIND(I, J)=X(I, J)
WIND(I+1, J)=Y(I, J)
GO TO 110
IF(I.EQ.IX)GC 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

```

Manstrath, John J.
 Storm surge simulation in transformed coordinates / by John J. Manstrath, Robert E. Whitaker...[et al.]. - Fort Belvoir, Va. : U.S. Coastal Engineering Research Center, 1976.
 2 v. : ill. (Technical report - U.S. Coastal Engineering Research Center ; no. 76-3) (Contract - U.S. Coastal Engineering Research Center ; DACW72-73-C-0014)
 CONTENTS : v.1. Theory and application. - v.2. Program documentation. Includes bibliographies.
 Report discusses a two-dimensional time-dependent numerical storm surge model using orthogonal curvilinear coordinates. Model is used in simulating storm surge induced by selected hurricanes.
 1. Storm surges. 2. Computer programs. 3. Hurricane Camille, 1969. 4. Hurricane Carla, 1961. 5. Hurricane Gracie, 1959. I. Title. II. Whitaker, Robert E., joint author. III. Series : U.S. Coastal Engineering Research Center. Contract DACW72-73-C-0014.

TC203 .U581tr no. 76-3 627 .U581tr

Manstrath, John J.
 Storm surge simulation in transformed coordinates / by John J. Manstrath, Robert E. Whitaker...[et al.]. - Fort Belvoir, Va. : U.S. Coastal Engineering Research Center, 1976.
 2 v. : ill. (Technical report - U.S. Coastal Engineering Research Center ; no. 76-3) (Contract - U.S. Coastal Engineering Research Center ; DACW72-73-C-0014)
 CONTENTS : v.1. Theory and application. - v.2. Program documentation. Includes bibliographies.
 Report discusses a two-dimensional time-dependent numerical storm surge model using orthogonal curvilinear coordinates. Model is used in simulating storm surge induced by selected hurricanes.
 1. Storm surges. 2. Computer programs. 3. Hurricane Camille, 1969. 4. Hurricane Carla, 1961. 5. Hurricane Gracie, 1959. I. Title. II. Whitaker, Robert E., joint author. III. Series : U.S. Coastal Engineering Research Center. Contract DACW72-73-C-0014.

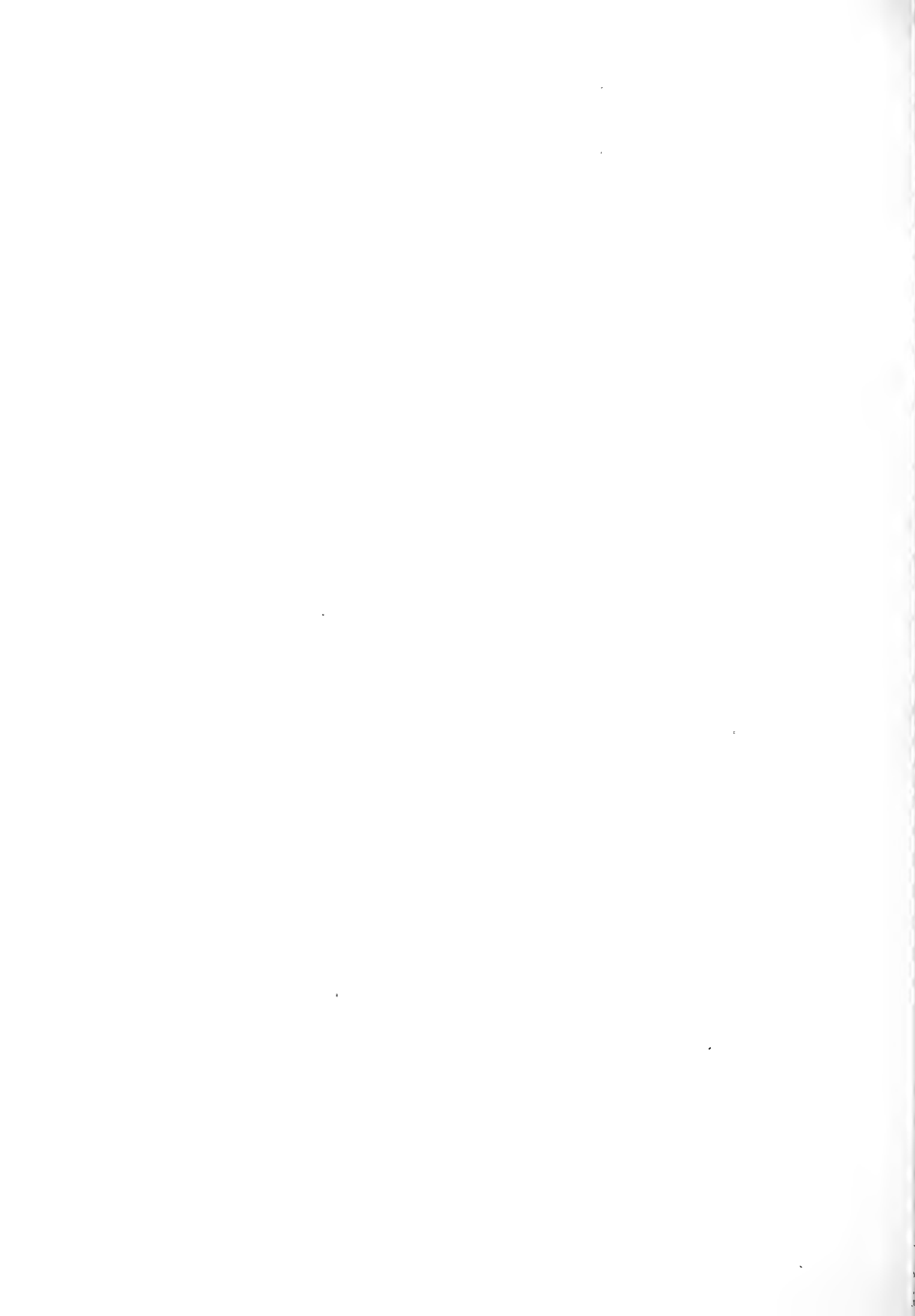
TC203 .U581tr no. 76-3 627 .U581tr

Manstrath, John J.
 Storm surge simulation in transformed coordinates / by John J. Manstrath, Robert E. Whitaker...[et al.]. - Fort Belvoir, Va. : U.S. Coastal Engineering Research Center, 1976.
 2 v. : ill. (Technical report - U.S. Coastal Engineering Research Center ; no. 76-3) (Contract - U.S. Coastal Engineering Research Center ; DACW72-73-C-0014)
 CONTENTS : v.1. Theory and application. - v.2. Program documentation. Includes bibliographies.
 Report discusses a two-dimensional time-dependent numerical storm surge model using orthogonal curvilinear coordinates. Model is used in simulating storm surge induced by selected hurricanes.
 1. Storm surges. 2. Computer programs. 3. Hurricane Camille, 1969. 4. Hurricane Carla, 1961. 5. Hurricane Gracie, 1959. I. Title. II. Whitaker, Robert E., joint author. III. Series : U.S. Coastal Engineering Research Center. Contract DACW72-73-C-0014.

TC203 .U581tr no. 76-3 627 .U581tr

Manstrath, John J.
 Storm surge simulation in transformed coordinates / by John J. Manstrath, Robert E. Whitaker...[et al.]. - Fort Belvoir, Va. : U.S. Coastal Engineering Research Center, 1976.
 2 v. : ill. (Technical report - U.S. Coastal Engineering Research Center ; no. 76-3) (Contract - U.S. Coastal Engineering Research Center ; DACW72-73-C-0014)
 CONTENTS : v.1. Theory and application. - v.2. Program documentation. Includes bibliographies.
 Report discusses a two-dimensional time-dependent numerical storm surge model using orthogonal curvilinear coordinates. Model is used in simulating storm surge induced by selected hurricanes.
 1. Storm surges. 2. Computer programs. 3. Hurricane Camille, 1969. 4. Hurricane Carla, 1961. 5. Hurricane Gracie, 1959. I. Title. II. Whitaker, Robert E., joint author. III. Series : U.S. Coastal Engineering Research Center. Contract DACW72-73-C-0014.

TC203 .U581tr no. 76-3 627 .U581tr



Manstrath, John J.
Storm surge simulation in transformed coordinates / by John J. Manstrath, Robert E. Whitaker...[et al.]. - Fort Belvoir, Va. : U.S. Coastal Engineering Research Center, 1976.
2 v. : ill. (Technical report - U.S. Coastal Engineering Research Center ; no. 76-3) (Contract - U.S. Coastal Engineering Research Center ; DACW72-73-C-0014)
CONTENTS : v.1. Theory and application. - v.2. Program documentation. Includes bibliographies.
Report discusses a two-dimensional time-dependent numerical storm surge model using orthogonal curvilinear coordinates. Model is used in simulating storm surge induced by selected hurricanes.
1. Storm surges. 2. Computer programs. 3. Hurricane Camille, 1969. 4. Hurricane Carla, 1961. 5. Hurricane Gracie, 1959. I. Title. II. Whitaker, Robert E., joint author. III. Series : U.S. Coastal Engineering Research Center. Contract DACW72-73-C-0014).

TC203 .U581tr no. 76-3 627 .U581tr

Manstrath, John J.
Storm surge simulation in transformed coordinates / by John J. Manstrath, Robert E. Whitaker...[et al.]. - Fort Belvoir, Va. : U.S. Coastal Engineering Research Center, 1976.
2 v. : ill. (Technical report - U.S. Coastal Engineering Research Center ; no. 76-3) (Contract - U.S. Coastal Engineering Research Center ; DACW72-73-C-0014)
CONTENTS : v.1. Theory and application. - v.2. Program documentation. Includes bibliographies.
Report discusses a two-dimensional time-dependent numerical storm surge model using orthogonal curvilinear coordinates. Model is used in simulating storm surge induced by selected hurricanes.
1. Storm surges. 2. Computer programs. 3. Hurricane Camille, 1969. 4. Hurricane Carla, 1961. 5. Hurricane Gracie, 1959. I. Title. II. Whitaker, Robert E., joint author. III. Series : U.S. Coastal Engineering Research Center. Contract DACW72-73-C-0014).

TC203 .U581tr no. 76-3 627 .U581tr

Manstrath, John J.
Storm surge simulation in transformed coordinates / by John J. Manstrath, Robert E. Whitaker...[et al.]. - Fort Belvoir, Va. : U.S. Coastal Engineering Research Center, 1976.
2 v. : ill. (Technical report - U.S. Coastal Engineering Research Center ; no. 76-3) (Contract - U.S. Coastal Engineering Research Center ; DACW72-73-C-0014)
CONTENTS : v.1. Theory and application. - v.2. Program documentation. Includes bibliographies.
Report discusses a two-dimensional time-dependent numerical storm surge model using orthogonal curvilinear coordinates. Model is used in simulating storm surge induced by selected hurricanes.
1. Storm surges. 2. Computer programs. 3. Hurricane Camille, 1969. 4. Hurricane Carla, 1961. 5. Hurricane Gracie, 1959. I. Title. II. Whitaker, Robert E., joint author. III. Series : U.S. Coastal Engineering Research Center. Contract DACW72-73-C-0014).

TC203 .U581tr no. 76-3 627 .U581tr

Manstrath, John J.
Storm surge simulation in transformed coordinates / by John J. Manstrath, Robert E. Whitaker...[et al.]. - Fort Belvoir, Va. : U.S. Coastal Engineering Research Center, 1976.
2 v. : ill. (Technical report - U.S. Coastal Engineering Research Center ; no. 76-3) (Contract - U.S. Coastal Engineering Research Center ; DACW72-73-C-0014)
CONTENTS : v.1. Theory and application. - v.2. Program documentation. Includes bibliographies.
Report discusses a two-dimensional time-dependent numerical storm surge model using orthogonal curvilinear coordinates. Model is used in simulating storm surge induced by selected hurricanes.
1. Storm surges. 2. Computer programs. 3. Hurricane Camille, 1969. 4. Hurricane Carla, 1961. 5. Hurricane Gracie, 1959. I. Title. II. Whitaker, Robert E., joint author. III. Series : U.S. Coastal Engineering Research Center. Contract DACW72-73-C-0014).

TC203 .U581tr no. 76-3 627 .U581tr

