REPORT NO. DOT-SLS-63-78-1,III

SAINT LAWRENCE SEAWAY NAVIGATION-AID SYSTEM STUDY
Volume III - Appendix C - User's Manual and Documentation of the Ship Maneuvering Requirements Computer Program

Jack W. Lewis
John J. Nelka

ARCTEC Inc.
9104 Red Branch Road
Columbia MD 21045

SEPTEMBER 1978
FINAL REPORT

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
ST. LAWRENCE SEAWAY DEVELOPMENT CORPORATION
Office of Comprehensive Planning
Washington DC 20591

Technical Report Documentation Page

1. Report No. DOT-SLS-63-78-1,III	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle SAINT LAWRENCE SEAWAY NAVIGATION-AID SYSTEM STUDY Volume III - Appendix C, User's Manual and Documentation of the Ship Maneuvering Requirements Computer Program		5. Report Date September 1978
		6. Performing Organization Code
7. Author(s) John W. Lewis, John J. Nelka		8. Performing Organization Report No. DOT-TSC-SLSDC-78-1,III
9. Performing Organization Name and Address ARCTEC, Inc. 9104 Red Branch Road Columbia MD 21045*		10. Work Unit No. (TRAIS) SL801/R8901
		11. Contract or Grant No. DOT-TSC-1395-III
12. Sponsoring Agency Name and Address U.S. Department of Transportation St. Lawrence Seaway Development Corporation Office of Comprehensive Planning Washington DC 20591		13. Type of Report and Period Covered Final Report Aug. 1977 - July 1978
		14. Sponsoring Agency Code
15. Supplementary Notes *Under contract to: U.S. Department of Transportation Research and Special Programs Administration Transportation Systems Center Cambridge MA 02142		

16. Abstract

The requirements for a navigation guidance system which will effect an increase in the ship processing capacity of the Saint Lawrence Seaway (Lake Ontario to Montreal, Quebec) are developed. The requirements include a specification of system positioning accuracy and the type and frequency of information which must be displayed to the master of each ship in the Seaway. A detailed development of the logic used to compute Seaway capacity as a function of the guidance system positioning accuracy is presented. A computer program is given which follows this logic and is used to compute Seaway capacity as a function of positioning accuracy for two classes of ships. Various sensitivity analyses are presented. It is shown that the capacity of the Seaway could be increased by up to 30 percent through the use of a navigation guidance system. Volume I, 116 pages, contains the main text and Appendixes A and D. Volume II, 128 pages, contains Appendix B.

17. Key Words St. Lawrence Seaway, Navigational Aids, Guidance Systems, All Weather Navigation, Electronic Navigation	18. Distribution Statement DOCUMENT IS AVAILABLE TO THE U.S. PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassifed	108	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F

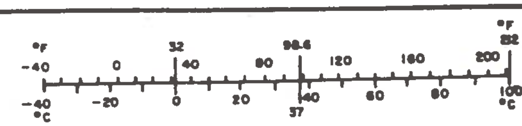


TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

a,b,c,d	Constants used in rudder control law
B	Ship beam
H/T	Ratio of water depth to ship draft
I_z	Ship mass moment of inertia about z-axis
L	Ship length
M	Ship maneuvering width
m	Ship mass
N	Moment about an axis through center of gravity of ship and parallel to z-axis
N_{wind}	Moment due to wind
u	Component of resultant velocity of the ship along the x-axis
$\dot{u}$	Component of resultant linear acceleration of the ship along the x-axis
v	Component of resultant velocity of the ship along the y-axis
$\dot{v}$	Component of resultant linear acceleration of the ship along the y-axis
W	Minimum channel width for reach
X	Longitudinal force
X_{wind}	Longitudinal force due to wind
$\overline{x}$	Center of action of Y_o, $\overline{x} = N_o/Y_oL$
x_{CG}	Longitudinal distance from origin to ship center of gravity
x_o	Position of ship along channel centerline
$\dot{x}_o$	Forward velocity of ship
$\dot{x}_{o\text{initial}}$	Initial equilibrium value of $\dot{x}_o$
$\ddot{x}_o$	Forward acceleration of ship ($\equiv$0.0)
Y	Side force
Y_{wind}	Side force due to wind

y_{course}	Desired lateral position in channel
y_o	Distance from channel centerline
$(y_o)_{CG}$	Maximum excursion of ship's center of gravity from desired course
$\dot{y}_o$	Lateral velocity of ship
$\dot{y}_{o current}$	Cross current velocity
$\ddot{y}_o$	Lateral acceleration of ship
δ	Angular displacement of rudder
ε	Navigation position accuracy
α	Correction factor to Y_o'
ϕ	Yaw angle
$\phi_{desired}$	Desired ship yaw angle
ϕ_{ss}	Yaw angle from steady state equations of motion
$\dot{\phi}$	Yaw angular velocity of ship
$\ddot{\phi}$	Yaw angular acceleration of ship

<u>Subscripts</u>

u	Indicates partial derivative with respect to u
$\dot{u}$	Indicates partial derivative with respect to $\dot{u}$
v	Indicates partial derivative with respect to v
$\dot{v}$	Indicates partial derivative with respect to $\dot{v}$
y_o	Indicates partial derivative with respect to y_o
δ	Indicates partial derivative with respect to δ
ϕ	Indicates partial derivative with respect to ϕ
$\dot{\phi}$	Indicates partial derivative with respect to $\dot{\phi}$
$\ddot{\phi}$	Indicates partial derivative with respect to $\ddot{\phi}$

C.1 GENERAL INFORMATION

C.1.1 Summary

A ship maneuvering simulation computer program (MANVER) was developed to determine the navigational guidance system performance requirements which would allow improvement in the capacity of the St. Lawrence Seaway. The maneuvering simulation provided the relationship between ship maneuvering room requirements and the weather conditions, and Seaway reach parameters. Solution to the controlled equations of motion was obtained using the Runge-Kutta fourth order integration technique.

Input to program MANVER consists of ship characteristics, which include linearized hydrodynamic force characteristics, ship geometry, and ship speed. Wind and current data, plus channel geometry, are also required as input.

Output of program MANVER is a time history of the ship's position and orientation in a Seaway reach. The maximum excursion of the ship from its desired path and the maximum rudder angle determine the ship maneuvering room requirements.

Program MANVER was written in the Fortran computer language and was developed using a Honeywell 1648 Time Sharing computer system. MANVER requires approximately 15 seconds connect time to compile, and from 5 to 60 minutes connect time to perform one computer simulation run of a ship traversing a reach at a given speed with given WX conditions. This execution time depends on the input data, channel length, and the time increment used to update variables.

C.1.2 Environment

A. User Organizations:

Transportation Systems Center
Kendall Square
Cambridge, Massachusetts 02142

St. Lawrence Seaway Development Corporation
800 Independence Avenue, SW
Federal Office Building #10, Room 836F
Washington, D.C. 20591

B. Program was developed using the Honeywell 1648 Time Sharing Computer System located at:

Dialcom, Inc.
1104 Spring Street
Silver Spring, Maryland 20910

C.1.3 References

1. Lewis, J.W., et al., "Saint Lawrence Seaway System Plan for All-Year Navigation (SPAN)," ARCTEC, Incorporated, July 1975.

2. Tippetts-Abbett-McCarthy-Stratton, "Study of Stability of Ice Cover-Phase II, Hydraulics Under Ice-Free Conditions," October 1972.

3. Grumblatt, J. L., "Great Lakes Water Temperatures, 1966-1975," Great Lakes Environmental Research Laboratory, NOAA Technical Memorandum ERL-GRERL-11

4. Majewski, W., et al., "A Study of the Thermal Balance of the Saint Lawrence River by Digital Simulations," National Research Council, Division of Mechanical Engineering, Report LTR-HY-5, March 1970.

5. Principles of Naval Architecture, Revised, Third Printing 1974, Society of Naval Architects and Marine Engineers.

6. Farwell, Capt. R. F., "Farwell's Rules of the Nautical Road," Naval Institute Press, Annapolis, Maryland, Fifth Edition.

7. FIPS Publication 38, "Guidelines for Documentation of Computer Programs and Automated Data Systems," February 1976.

C.2 APPLICATION

C.2.1 Description of Program MANVER

A functional relationship between ship maneuvering room requirements, the weather conditions, and the Seaway reach parameters was determined using the equation of ship motion to describe the movement of a ship when subjected to external disturbance forces (winds, currents, bank suction effects) and control forces (rudder and propeller). The equations of motion and a ship control equation are presented below.

$$(m - X_{\dot{u}})\ddot{x}_o = X_{wind} + X_\delta \delta + X_{y_o} y_o + (\dot{x}_o - \dot{x}_{o\,initial}) X_u \qquad (1)$$

$$(m - Y_{\dot{v}})\ddot{y}_o = Y_{wind} + Y_\delta \delta + Y_{y_o} y_o + (\dot{y}_o - \dot{y}_{o\,current}) Y_v - Y_v(\dot{x}_o - \dot{x}_{o\,current})\phi + (Y_{\dot{\phi}} - Y_{\dot{v}}\dot{x}_o)\dot{\phi} + (Y_{\ddot{\phi}} - m_{CG})\ddot{\phi} \qquad (2)$$

$$(I_z - N_{\ddot{\phi}})\ddot{\phi} = N_{wind} + N_\delta \delta + N_{y_o} y_o + N_v(\dot{y}_o - \dot{y}_{o\,current}) + (N_{\dot{v}} - m_{CG})\ddot{y}_o + N_v(\dot{x}_o - \dot{x}_{o\,current})\phi + (N_{\dot{\phi}} - N_{\dot{v}}\dot{x}_o)\dot{\phi} \qquad (3)$$

$$\delta = a(y_o - y_{course}) + b\dot{y}_o + c(\phi - \phi_{desired}) + d\dot{\phi} \qquad (4)$$

Ship coordinate system and motion variables are described in Figure C.1. Figure C.2 presents a ship in a channel at its initial position ① and during a simulation. The initial yaw angle and rudder are determined from the steady state equations of motion using steady wind and current conditions. Position ② motion variables are determined from the unsteady equations of ship motion using unsteady wind and current data of the form shown in Figure C.3. The wind data is a function of time while the current is a function of position. The rudder angle plot shown is a result of the ship maneuvering simulation. A sketch of ship position and orientation obtained from a ship maneuvering simulation run is also shown in Figure C.3.

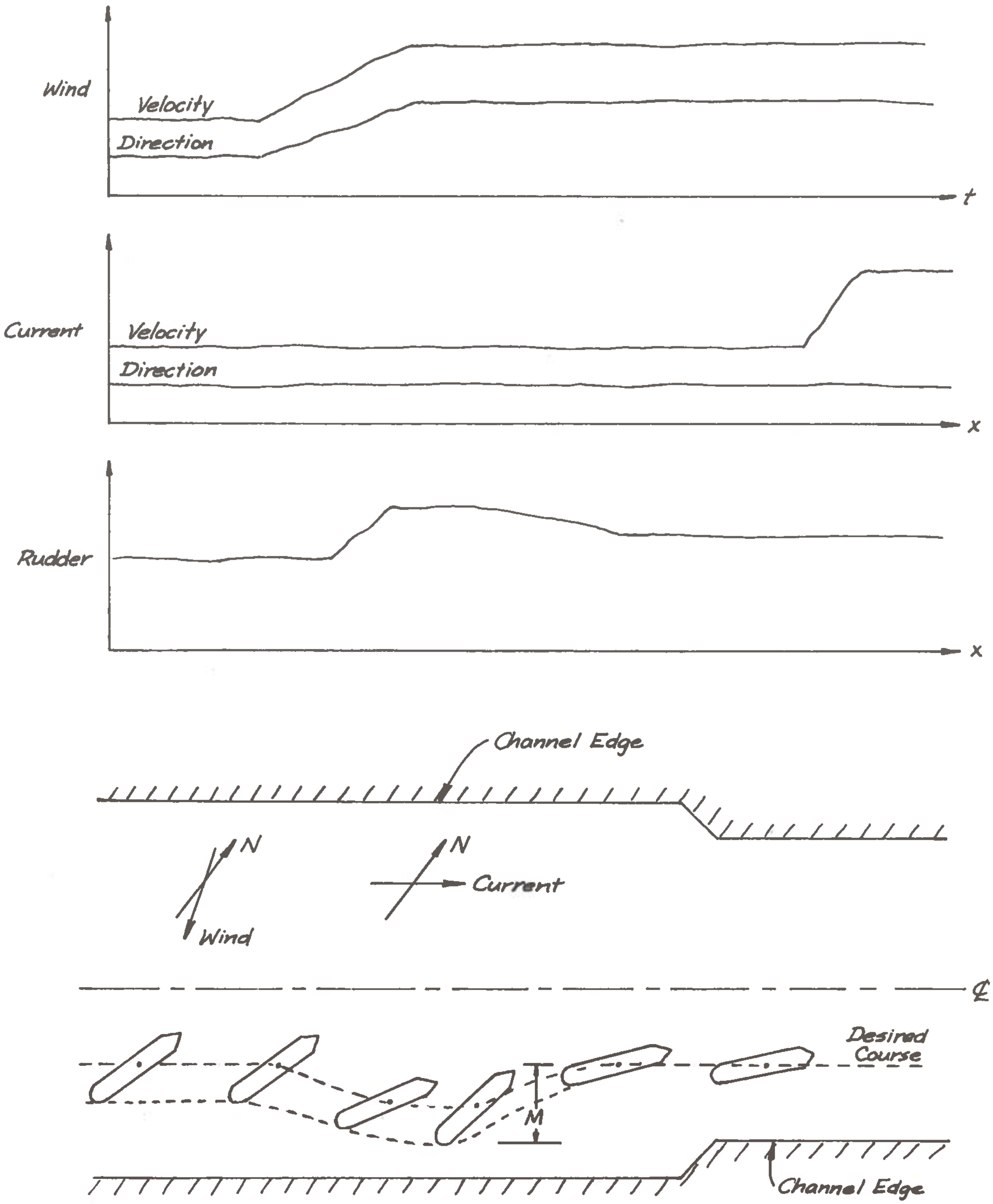


Figure C-3. Wind, Current and Rudder Profiles and a Sketch of Ship Position and Orientation during a Maneuvering Simulation

A block diagram of the ship maneuvering simulation, Figure C.4, illustrates the logic of program MANVER. The wind and current forces are determined for time t and entered into the numerical integrator along with ship heading and rudder forces and moments. The fourth order Runge-Kutta numerical integration is used to determine the following motion parameters with time t:

$$x_o,\ \dot{x}_o,\ \ddot{x}_o,\ y_o,\ \dot{y}_o,\ \ddot{y}_o,\ \phi,\ \dot{\phi},\ \ddot{\phi}$$

For the control law employed in this study, the motion variables--y_o, $\dot{y}_o$, ϕ,$\dot{\phi}$--would then determine the rudder angle for time $(t+\Delta t)$. The updated rudder force and moment are then computed for time $(t+\Delta t)$ and are entered into the numerical integrator along with the updated wind and current forces and moments and ship heading. This procedure is repeated until the ship transits to the end of the channel. The simulation determines rudder angle, ship heading, $\dot{y}_o$, $\dot{\phi}$, and the maneuvering room (width) requirement, M.

The maneuvering width, M, is defined to be

$$M = (y_o)_{CG} + \frac{1}{2}(L\sin\phi + B\cos\phi)$$

where

$(y_o)_{CG}$ = maximum excursion of ship center of gravity from the desired path

L = ship length

B = ship beam

ϕ = ship yaw angle

The ship maneuvering simulation was used to determine the maneuvering width for three Seaway reaches as a function of beam wind speed and ship speed. These results for reaches 44 (along St. Regis Island), 50 (Polly's Gut), and 56 (Copeland Cut), are tabulated in Tables C.1, C.2, and C.3, respectively.

C.2.2 Equipment

Program MANVER is designed to run on a Honeywell 1648. The program is written in Fortran IV as used by the compiler on the Dialcom time sharing system. The major differences between this compiler and standard Fortran are:

(1) Comments are denoted by *.

(2) Continuations are denoted by +.

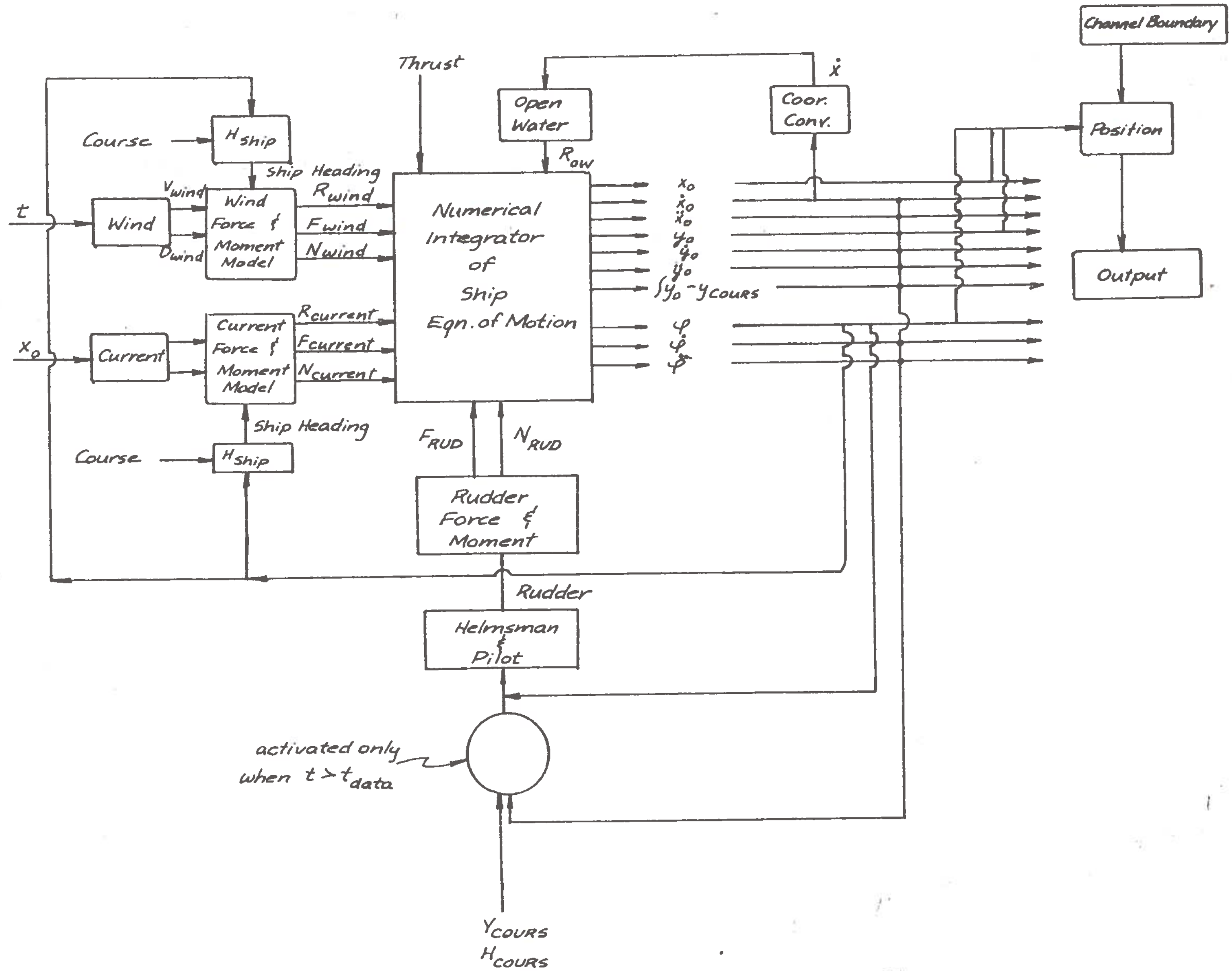


Figure C.4. Block Diagram of Ship Maneuvering Simulation

TABLE C.1 DYNAMIC MANEUVERING RESULTS FOR SEAWAY REACH 44 (ALONG ST. REGIS ISLAND) FOR A SALTY AND A LAKER

Wind Direction (°T)	Wind Speed (mph)	Ship Speed (mph)	Salty Upbound *M* (ft)	Salty Downbound *M* (ft)	Laker Upbound *M* (ft)	Laker Downbound *M* (ft)
331.17	0	3	819.10	1,331.40	819.78	1,334.21
		6	464.05	627.47	464.32	635.03
		9	316.57	394.74	317.64	399.18
		12	249.49	285.74	250.19	288.67
	15	3	831.82	1,379.06	834.29	1,408.58
		6	466.97	641.85	467.24	651.12
		9	317.59	397.89	318.39	402.76
		12	249.93	286.79	250.48	289.93
	30	3	865.13	1,138.14	903.10	1,200.84
		6	475.98	690.23	478.32	706.74
		9	321.99	412.91	322.26	419.77
		12	252.46	293.18	252.57	297.34
	45	3	30,457.00	3,080.05	∞	∞
		6	490.90	1,220.61	496.77	1,229.00
		9	329.67	443.07	331.79	454.86
		12	257.08	304.82	258.32	310.80
151.17	15	3	824.00	1,981.75	825.70	∞
		6	462.98	606.08	464.08	611.40
		9	316.82	387.89	318.28	391.44
		12	248.29	282.67	249.21	285.20
	30	3	5,500.70	3,217.50	∞	∞
		6	475.64	1,397.76	479.10	1,400.01
		9	322.55	369.35	325.19	370.41
		12	250.60	273.96	252.23	275.27
	45	3	35,760.30	∞	∞	∞
		6	751.47	3,207.50	771.30	3,223.50
		9	332.01	795.46	336.63	811.39
		12	256.09	271.24	258.90	273.25

TABLE C.2 DYNAMIC MANEUVERING RESULTS FOR SEAWAY REACH 50 (POLLY'S GUT) FOR A SALTY AND A LAKER

Wind Direction (°T)	Wind Speed (mph)	Ship Speed (mph)	Salty Upbound M (ft)	Salty Downbound M (ft)	Laker Upbound M (ft)	Laker Downbound M (ft)
357	0	3	452.21	617.72	454.22	627.56
		6	317.73	350.86	321.32	355.44
		9	232.04	248.84	235.63	251.35
		12	183.27	192.98	185.58	194.72
	15	3	463.83	1,070.65	471.26	1,087.10
		6	321.31	402.14	325.46	408.29
		9	233.49	262.64	237.03	265.88
		12	183.87	198.16	187.41	200.11
	30	3	489.12	5,512.92	499.20	5,525.00
		6	331.76	600.08	338.25	612.71
		9	238,74	315.52	242.91	321.02
		12	287.06	219.95	189.90	223.08
	45	3	3,277.64	17,079.67	3,295.14	17,200.00
		6	351.35	3,227.82	374.06	3,300.00
		9	248.37	412.55	253.62	421.95
		12	192.77	258.67	196.34	263.70
177	15	3	453.26	255.47	459.08	259.99
		6	315.40	294.71	318.56	296.87
		9	230.34	232.41	233.49	234.37
		12	182.21	186.38	184.39	187.81
	30	3	443.64	5,686.79	446.96	5,700.00
		6	306.95	180.07	308.92	184.36
		9	224.92	187.38	226.11	188.28
		12	178.89	163.91	180.59	164.10
	45	3	14,804.23	17,260.94	14,850.00	17,300.00
		6	297.82	3,974.76	298.80	4,000.00
		9	218.33	142.35	218.62	146.77
		12	173.45	143.92	174.37	148.34

TABLE C.3 DYNAMIC MANEUVERING RESULTS FOR SEAWAY REACH 56 (COPELAND CUT) FOR A SALTY AND A LAKER

			Salty		Laker	
Wind Direction (°T)	Wind Speed (mph)	Ship Speed (mph)	Upbound *M* (ft)	Downbound *M* (ft)	Upbound *M* (ft)	Downbound *M* (ft)
347	0	3	225.87	225.62	228.05	228.24
		6	225.26	226.44	230.59	232.55
		9	201.67	211.74	252.49	217.85
		12	144.71	147.28	213.19	153.75
		15	117.00	118.59	126.44	122.49
	15	3	225.27	225.30	228.24	249.83
		6	226.26	226.36	230.39	230.12
		9	197.75	204.92	252.56	214.73
		12	142.64	144.83	203.20	150.89
		15	115.20	117.46	115.99	121.63
	30	3	225.47	267.29	261.52	265.00
		6	225.27	225.07	226.76	237.71
		9	189.57	186.78	249.73	193.62
		12	137.04	137.31	172.26	141.94
		15	111.83	113.56	112.12	116.98
	45	3	225.03	227.33	260.91	235.00
		6	226.88	226.09	243.29	258.12
		9	214.39	225.78	253.84	229.97
		12	129.02	126.09	131.41	128.50
		15	106.75	107.35	107.34	109.50
167	15	3	225.71	225.46	231.67	242.50
		6	229.17	225.36	235.26	231.33
		9	210.16	215.88	293.90	227.39
		12	190.00	148.88	234.29	155.68
		15	179.02	119.34	187.18	123.89
	30	3	227.38	267.29	232.08	270.00
		6	230.04	225.45	238.75	236.84
		9	217.17	225.04	314.14	237.34
		12	205.00	156.54	252.93	164.73
		15	193.61	123.07	203.77	128.36
	45	3	229.67	300.76	264.99	350.00
		6	228.91	225.66	246.68	253.01
		9	228.71	225.34	245.72	237.94
		12	224.00	172.16	264.91	182.86
		15	220.34	130.41	233.99	137.04

(3) The end-of-file marker assigns a value of 1 to variable IEOF.

(4) A series of statements separated by a semi-colon (;) may be placed on the same line and all will be carried out. If the series follows a decision, the entire series will be executed or not executed.

(5) Line numbers occupy the first five columns of each line.

C.2.3 Structure of Program MANVER

Figure C.5 shows the calling relationships between the main program and the various subroutines. The main program, MANVER, keeps track of the simulation time to update and print the variables as needed. Subroutine INDAT reads the input data from two data files and prints the data onto the output file. To print the current date in the heading, INDAT calls DATE, a system subroutine not contained in the program. Subroutine SSCON is called by MANVER to establish steady-state conditions for the ship and the initial time and distance. A linear interpolation of the course, edge, current, and wind arrays are provided by subroutine INTER. The frontal and lateral forces and the yaw moment due to wind are calculated by WIND. Subroutine RESIST calculates the ship's open water resistance. Subroutine CHCOEF is an interpolation algorithm that determines forces and moments acting on ships operating in shallow waters and narrow channels. As the time increments in the simulation, subroutine SOLVE integrates the differential equations by the Runge-Kutta method using the derivatives for ship motion obtained in subroutine CALDER. The rudder commands of a pilot and helmsman are simulated by subroutine RUDDER. As directed by the main program, subroutine PRT prints the changing values of the simulation.

C.2.4 Performance

Every case requires input files SIMDTA and ADSIM and generates an output file OUT384. If more than one case is being run, the SIMDTA file must contain one full set of data for each case. ADSIM is not lengthened but rewound at the beginning of each case. Output from more than one case will be added to the output file without special markers, other than normal headings, between cases. The running time for each case varies from 5 to 60 minutes, depending on the input data, especially ship speed, distance to be covered, and time increment for updating variables.

C.2.5 Data Base

Three data files are maintained by this program, two input and one output. Input file SIMDTA contains information on the simulation controls, desired course, channel boundaries, wind and current data, maneuvering control coefficients, and ship data. This file is read from beginning to end with the end-of-file signaling the end of the run. Input file ADSIM contains information on the dimensionless side force Y_o' as a function of channel width, ship beam, and distance from channel centerline. It is read in order

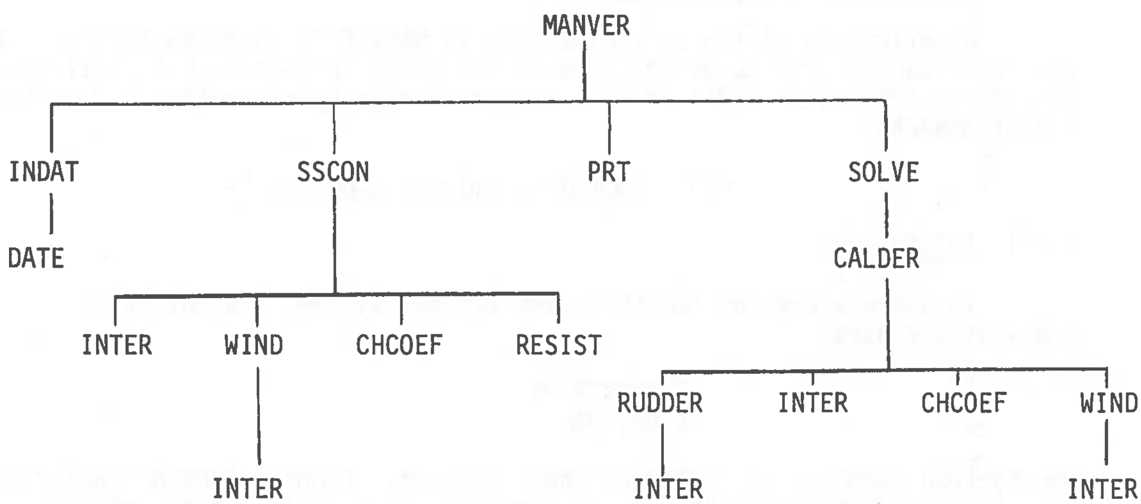


FIGURE C.5 STRUCTURE OF PROGRAM MANVER

but may be rewound and read several times, depending on the number of cases in the run. Output file OUT384 is generated by each run. It contains a listing of all the input data read from files SIMDTA and ADSIM for the case and the simulation results. If more than one case is to be run, the output is written in the order generated.

C.2.6 Input, Processing, and Output

A definition of the variables used in MANVER is given in Section C.4 and detailed flow charts of the program are given in Section C.5. All input is done by subroutine INDAT and all output is made by subroutine PRT at the end of MANVER.

C.3 PROCEDURES AND REQUIREMENTS

C.3.1 Initiation

To execute program MANVER on the Dialcom system, the following commands are used:

```
F MANVER MA
LLOAD MA
```

The Fortran compiler is called by the F command. Program MANVER, including all of the attached subroutines, is compiled and an object file, MA, is created. The LLOAD command loads the object file and execution begins.

C.3.2 Input

All data is input from files SIMDTA and ADSIM (see Tables C.4 and C.5). The format statements referenced indicate the maximum number of characters (including separating comma) that the variable may contain. The formats are used to print the data in an understandable and orderly manner while the input is essentially free format. The two data files will be described line-by-line with the numbers corresponding to the position in the sample file; possible differences will be noted.

File SIMDTA

1 - run number for heading (2A2)
2 - description of following line (30A2)
3 - RHOA,RHOW,GRAV,DISPLC(4F10.2)
4 - description (30A2)
5 - TSTART,XSTART,XEND(3F10.2)
6 - description (30A2)
7 - DTSIM,DTPRT,DTDATA(3F10.2)
8 - description (30A2)
9 - NCOURS,NEDGE,NCURNT,WDEPTH(3I5,F10.2)
10 - description (30A2)
11-13- COURSE(I,J),J=1,3 where I increments with each line from 1 to NCOURS, maximum of 10 (3F10.2)
14 - description (30A2)

TABLE C.4 LISTING OF INPUT FILE SIMDTA

```
1
      RHOA       RHOW     GRAVITY    #/LT
0.00237,1.94,32.2,2240.
     TSTART   XSTART   XEND
0.0,-500.0,10000.0
     DTSIM      DTPRT      DTDATA
10.0,60.,0.1
NCOURS NEDGE NCURNT WDEPTH
3,9,7,30.0
      XO        YCOURS    HEADING
-500.0,-15.0,267.0
5000.0,-15.0,267.0
10000.0,-15.0,267.0
      XO     CHANNEL BOUNDARY
-500.0,350.0,-350.0
0.0,350.0,-350.0
1000.0,375.0,-375.0
2000.0,400.0,-400.0
3000.0,450.0,-450.0
4000.0,500.0,-500.0
6500.0,500.0,-500.0
7000.0,500.0,-500.0
10000.0,500.0,-500.0
      XO VCURRENT(FPS) DIRECTION
0.0,6.5,87.0
1000.0,6.5,87.0
2000.0,6.5,91.0
3000.0,6.5,94.0
4000.0,6.5,97.0
5000.0,0.0,97.0
10000.0,0.0,97.0
           WIND DATA
4
     TIME  VWIND(MPH)  DIRECTION
0.0,0.0,357.0
20000.,0.0,357.
21000.,0.0,357.
90000.0,0.0,357.0
                     CPOS ARRAY
0.0005,0.04,0.,0.0000000000001
                    CHEAD ARRAY
2.,287.,0.
          NON-DIMENSIONAL MANEUVERING COEFFICIENTS
-0.0067,-0.0106,00.0,-0.0004,0.0029
-0.0004,0.0003,-0.0022,-0.0035,0.0
0.0012,-0.0006,00.0,0.0,0.0
DISPLACEMENT   BEAM      DRAFT            XIZ
22932.0,75.0,25.0,534000000000.
     XLBP      XLWL       XLOA
565.0,565.0,600.0
     VSHIP  TSHIP       AREAP     AREAF
10.0,170500.0,11300.0,0.0
MAX RUDDER (INPUT IN DEG, PRINT IN RAD)
35.0
```

TABLE C.5 LISTING OF INPUT FILE ADSIM

```
     NYO' AND YYO' DATA
  NWPB NXALP
7,11
               XWPB ARRAY
2.0,3.0,4.0,5.0,6.0,7.0,8.0
               NYPB ARRAY
6,6,9,11,8,9,10
XYPO ARRAY, READ 6 ENTRIES PER LINE, FIRST YO/B THEN YO' LINE
0.0,0.1,0.2,0.3,0.4,0.52
0.0,0.005,0.019,0.036,0.062,0.102
0.0,0.2,0.4,0.6,0.8,1.0
0.0,0.008,0.019,0.030,0.056,0.091
0.0,0.2,0.4,0.6,0.8,1.0
1.2,1.4,1.45
0.0,0.004,0.008,0.014,0.022,0.033
0.048,0.070,0.080
0.0,0.2,0.4,0.6,0.8,1.0
1.2,1.4,1.6,1.8,2.0
0.0,0.003,0.006,0.010,0.013,0.018
0.022,0.030,0.040,0.050,0.067
0.0,0.4,0.8,1.2,1.6,2.0
2.4,2.5
0.0,0.008,0.010,0.016,0.022,0.034
0.052,0.056
0.0,0.4,0.8,1.2,1.6,2.0
2.4,2.8,3.0
0.0,0.007,0.009,0.012,0.017,0.021
0.028,0.039,0.045
0.0,0.4,0.8,1.2,1.6,2.0
2.4,2.8,3.2,3.5
0.0,0.006,0.008,0.010,0.014,0.016
0.018,0.022,0.030,0.035
              XALPHA ARRAY
      H/T          XBAR      ALPHA
1.0,-0.50,1.30
1.2,-0.35,1.10
1.4,-0.28,1.00
1.6,-0.23,0.92
1.8,-0.20,0.86
2.0,-0.17,0.80
2.2,-0.15,0.76
2.4,-0.13,0.70
2.6,-0.11,0.67
2.8,-0.10,0.62
3.0,-0.09,0.59
```

?

15-23- EDGE(I,J),J=1,3 where I increments with each line from 1 to NEDGE, maximum of 10 (3F10.2)

24 - description (30A2)

25-31- XCURNT(I,J),J=1,3 where I increments with each line from 1 to NCURNT, maximum of 10 (3F10.2)

32 - description (30A2)

33 - NWIND(I2)

34 - description (30A2)

35-38- XWIND(I,J),J=1,3 where I increments with each line from 1 to NWIND, maximum of 10 (3F10.2)

39 - description (30A2)

40 - CPOS(I),I=1,4 (3F12.7,F15.13)

41 - description (30A2)

42 - CHEAD(I),I=1,3 (3F12.7)

43 - description (30A2)

44-46- XMCOEF(I),I=1,15 (5F12.5)

47 - description (30A2)

48 - DISPL,BEAM,DRAFT,XIZ (3F10.2,F20.0)

49 - description (30A2)

50 - XLBP,XLWL,XLOA (3F10.2)

51 - description (30A2)

52 - VSHIP,TSHIP,AREAP,AREAF (4F10.2)

53 - description (30A2)

54 - RUDMAX (F10.2)

File ADSIM

1 - description of following line (30A2)

2 - description (30A2)

3 - NWPB,NXALP (2I5)

4 - description (30A2)

5 - XWPB(I),I=1,NWPB maximum of 7 (7F10.2)

6 - description (30A2)

7 - NYPB(I),I=1,NWPB maximum of 7 (7I5)

8 - description (30A2)

9-32- XYOP(I,J,K) (6F10.3)
This array is dimensioned (7,11,2) where the first dimension (I) is set by NWPB and the second dimension (J) is set by the corresponding I value of NYPB. The array is read in sets, from 1 to NWPB where each set contains

XYOP(I,J,1),J=1,NYPB(I) (6F10.2)
XYOP(I,J,2),J=1,NYPB(I) (6F10.2)

In sets where NYPB(I) is 6 or less, there are two lines; in sets where NYPB(I) is between 7 and 11, there are four lines

33 - description (30A2)

34-44- XALPHA(I,J),J=1,3 where I increments with each line from 1 to NXALP, maximum of 11 (3F10.2)

C.3.3 Output

The program output is written onto a file, OUT384, which can be disposed to a high speed printer or printed at the terminal. If the printing device requires that the first column contain only printer commands, all format statements must be shifted one space. This will cause the format in subroutine PRT to have 81 characters, all others will be 80 or less.

The output is divided into two sections, input data and simulation results. The input data section, printed by subroutine INDAT, is printed first and has a heading giving the date of the run and the run number. As data is read from the input files, it is printed on the output file in the same format. These formats are described in the input section of this guide. The headings for the result section are also printed in subroutine INDAT although the data are printed by subroutine PRT. This data is presented in columns and contain (from left to right):

Simulated time	SAMP(1)
Location in channel	SAMP(2)
Position off centerline	SAMP(6)
Ship heading	SAMP(9)
$\dot{x}_o$	SAMP(3)
$\dot{y}_o$	SAMP(7)
$\dot{\phi}$	SAMP(10)
Rudder angle	SAMP(12)
Excursion from channel centerline	SAMP(16)

The final line of output, printed by MANVER, gives the ship's maximum excursion and the location at which it occurred. A sample output is shown in Table C.6.

C.3.4 Error Recovery

Error messages may be produced by two of the subroutines, SSCON and CHCOEF. If any of these messages appear, the user should interrupt the program and check his input data for unusual numbers.

ERROR: SEARCH FOR YO GOING WRONG WAY TWICE. In SSCON, assumed and calculated values of YO are being compared to the desired set of numbers. This error occurs when neither increasing nor decreasing the assumed values approach the desired value yet the desired value is not bracketed by the assumptions. The user should check the course and current arrays for errors.

TABLE C.6 OUTPUT OF MANEUVERING SIMULATION

SIMULATION RESULTS

TIME (SEC)	XO (FT)	YO (FT)	SHIP HEADING (DEG)	DXO/DT (FPS)	DYO/DT (FPS)	DPSI/DT (RAD/SEC)	RUDDER (DEG)	DIST FR CL (FT)
0.00	-500.00	-15.00	267.00	14.67	0.00	0.00	0.00	52.50
60.00	380.20	-15.00	267.00	14.67	-0.00	-0.00	-0.00	52.50
120.00	1260.40	-15.16	266.99	14.67	-0.03	-0.00	-0.09	52.69
180.00	2140.60	-26.59	266.76	14.67	-0.42	-0.01	-2.86	65.35
240.00	3020.80	-65.43	266.65	14.67	-0.85	0.00	-3.54	104.77
300.00	3901.00	-123.60	267.03	14.67	-1.06	0.01	-2.79	161.24
360.00	4781.20	-176.86	268.11	14.67	-0.45	0.03	3.02	220.17
420.00	5661.40	-166.60	269.51	14.67	0.63	0.01	6.00	217.21
480.00	6541.61	-123.24	269.43	14.67	0.71	-0.01	1.23	173.42
540.00	7421.81	-86.59	268.80	14.67	0.51	-0.01	-0.21	133.49
600.00	8302.01	-61.27	268.26	14.67	0.35	-0.01	-0.17	105.38
660.00	9182.20	-43.72	267.88	14.67	0.24	-0.01	-0.07	85.83
720.00	10062.40	-31.55	267.60	14.67	0.17	-0.00	-0.05	72.19

MAX. EXCURSION FROM CHANNEL CENTERLINE = 227.83 AT XO = 5221.30

ERROR IN SSCON: PSI = __________ The difference between the ship heading and the course heading is greater than 90°. This difference, in radians, is printed.

ERROR IN SUBROUTINE CHCOEF AT STATEMENT 50 AYOPB OF __ LESS THAN __ The ratio of distance off center to ship beam (AYOPB) is less than the initial value of XYOP(L,1,1) which is zero. The values of AYOPB and XYOP(L,9,1) are printed. The user should check the XYOP array.

ERROR IN SUBROUTINE CHCOEF AT STATEMENT 40.

ERROR IN SUBROUTINE CHCOEF AT STATEMENT 90.

ERROR IN SUBROUTINE CHCOEF AT STATEMENT 300.

In each case, the do loop concluding at the statement number indicated has been incorrectly completed to the maximum number. The program should have left the loop by a statement within the loop. The user is advised to check the program with the listing to ensure that no alterations have been made. It is impossible for the program to go to these messages unless the program has been changed.

C.4 DEFINITION OF VARIABLES

ALPHA	draft/depth factor for lateral force due to restricted channel; interpolated from XALPHA array by subroutine CHCOEF
ANG	angle between ship heading and course (radians); calculated in subroutine SSCON
ANGLER	angle between ANG and steady state heading (radians); calculated in subroutine RUDDER
AREAF	ship frontal area; read from file SIMDTA by subroutine INDAT
AREAP	ship profile area; read from file SIMDTA by subroutine INPUT
AYOPB	ratio of the distance ship is off centerline of channel to ship beam; calculated in subroutine CHCOEF
BEAM	beam of ship (feet); read from file SIMDTA by subroutine INDAT
CE1	ships distance from positive channel boundary; interpolated from EDGE array by subroutine INTER
CE2	ships distance from negative channel boundary; interpolated from EDGE array by subroutine INTER
CHEAD(3)	rudder command controls related to ship heading; read from file SIMDTA by subroutine INDAT CHEAD(1) rudder command due to an error in ship yaw position from desired yaw position CHEAD(2) rudder command due to ship yaw velocity CHEAD(3) rudder command due to ship yaw acceleration
COURSE(10,3)	desired ship position and heading in channel; read from file SIMDTA by subroutine INDAT COURSE(N,1) x_o location of Nth point (feet); N=1, NCOURS COURSE(N,2) desired displacement from center of channel of Nth point (feet); N=1, NCOURS COURSE(N,3) desired ship heading of Nth point (°T); N=1, NCOURS

CPOS(4)	rudder command controls related to ship position; read from file SIMDTA by subroutine INDAT
	CPOS(1) rudder command due to an error in ship lateral position from desired lateral position
	CPOS(2) rudder command due to ship lateral velocity
	CPOS(3) rudder command due to ship lateral acceleration
	CPOS(4) rudder command due to integral of error in ship position
CSTVR(15)	dummy array of calculation variables; used in MANVER and subroutines SSCON, SOLVE, and CALDER
	1) XOD 6) PSID 11) YO 2) XO 7) HSHIPR 12) YOD 3) YOD 8) XODD 13) YODD 4) YO 9) YODD 14) YOI 5) YOI 10) PSIDD 15) XRUDDR
D1	intermediate step in calculation of PSI; used in subroutine SSCON
D2	intermediate step in calculation of PSI; used in subroutine SSCON
DD	time increment of one-half DTSIM used in Runge-Kutta estimate of ship movement; calculated in subroutine SOLVE
DELTA	percentage of change between points in array to interpolate for desired value; used in subroutine INTER
DHPT	difference between desired HPT value and value in XALPHA array; used in subroutine CHCOEF
DISPL	ship displacement (long tons); read from file SIMDTA by subroutine INDAT
DISPLC	conversion factor for long tons to pounds; read from file SIMDTA by subroutine INDAT
DRAFT	ship draft (feet); read from file SIMDTA by subroutine INDAT
DTAB1	value equal to XYOP(L,J,1)-XYOP(L,J-1,1); calculated in subroutine CHCOEF
DTAB2	value euqal to XYOP(L,J,2)-XYOP(L,J-1,2); calculated in subroutine CHCOEF
DTDATA	time increment for changing rudder (sec); read from file SIMDTA by subroutine INDAT

DTPRT	time increment for printing results (sec); read from file SIMDTA by subroutine INDAT
DTSIM	time increment for calculations (sec); read from file SIMDTA by subroutine INDAT
DWPB1	value equal to XWPB(I)-XWPB(I1); used in subroutine CHCOEF
DWPB2	difference between calculated value of WPB and array value at XWPB(I1); used in subroutine CHCOEF
DXAL1	value equal to XALPHA(K,1)-XALPHA(K-1,1); used in subroutine CHCOEF
DXAL2	value equal to XALPHA(K,2)-XALPHA(K-1,2); used in subroutine CHCOEF
CYO	increment for changing ship position; used in SSCON to calculate steady state conditions
DYOP	difference between values of YOP calculated from array for WPB value on either side of actual value, subroutine CHCOEF
DYOPB	value equal to YOP1-YOP2; used in subroutine CHCOEF
EDGE(10,3)	boundaries describing channel; read from file SIMDTA by subroutine INDAT
	EDGE(N,1) x_o location of Nth point (feet); N=1, NEDGE
	EDGE(N,2) right side channel boundary at Nth point (feet); N=1, NEDGE
	EDGE(N,3) left side channel boundary at Nth point (feet); N=1, NEDGE
FORCE	total wind force; used in subroutine WIND
FPSMPH	conversion for feet per second to miles per hour (1.467); initialized in MANVER
FWIND	lateral wind force; calculated in subroutine WIND
GRAV	acceleration due to gravity (ft/sec^2); read from file SIMDTA by subroutine INDAT
HCOURS	course heading at time of calculation (°T); interpolated from COURSE array by subroutine INTER

HCR	current heading at time of calculation (radians); used in subroutine SSCON and CALDER
HCSS	steady state course heading (degrees); calculated by SSCON
HCURNT	current heading at time of calculation (°T); interpolated from XCURNT array by subroutine INTER
HPT	ratio of depth of channel to ship draft; used in subroutine CHCOEF
HSHIP	ship heading at time of calculation (°T); used in subroutine SSCON and CALDER
HSHIPR	ship heading at time of calculation (radians); used in subroutine SSCON and CALDER
HSR	course heading at time of calculation (radians); used in subroutine SSCON and CALDER
HWIND	wind heading at time of calculation (°T); interpolated from XWIND array by subroutine INTER
HWINDR	wind heading at time of calculation (radians); used in subroutine WIND
I1	internal increment (I) minus 1, used as value L for array XYOP(L,J,K); calculated in subroutine CHCOEF
IA	code for calculating steady state rudder and PSI (1 - first calculated point above desired line, 2 - second calculated point below desired line); used in subroutine SSCON
IANG	code to indicate that third column of the array to be interpolated is an angle (=1); used in subroutine SSCON and CALDER
IB	code for calculating steady state rudder and PSI (1 - second calculated point above desired line, 2 - second calculated point below desired line); used in subroutine SSCON
IC	code for calculating steady state rudder and PSI (0 - both calculated points on same side of desired line, 1 - calculated points on opposite sides of desired line); used in subroutine SSCON
ID	code for calculating steady state rudder and PSI (0 - assumed initial point for second calculation greater than point for first, 1 - assumed initial point for second calculation less than point for first); used in subroutine SSCON

IDAY	day of run obtained from subroutine DATA (alphanumeric) and printed in heading; used in subroutine INDAT
IDPT	code to indicate if x_Q distance or time is increasing (1) or decreasing (-1) in array; used in subroutine INTER
IHEAD	code to determine if third column of the array to be interpolated is an angle (=1) for a distance (=0); used in subroutine INTER
IMON	month of run obtained from subroutine DATE (alphanumeric) and printed in heading; used in subroutine INDAT
IN	dummy variable for argument lists preceding input variables
IOUT	dummy variable for argument lists preceding output variables
IPT	increment variable for first of two dimensions in XYZ array, XYZ(IPT,1); used in subroutine INTER
ITITLE(30)	60 alphanumeric character lines describing lines of input data, read from files SIMDTA and ADSIM and printed on file OUT384 by subroutine INDAT
IYER	year of run obtained from subroutine DATE (alphanumeric) and printed in heading; used in subroutine INDAT
JPT	increment variable for first of two dimensions in XYZ array, XYZ(JPT,1); used in subroutine INTER
JT	number of observations in the last of three dimensions in XYOP array, XYOP(I,J,JT); used in subroutine INDAT
MYPB	number of observations in last of three dimensions in XYOP array, XYOP(I,J,MYPB); used in subroutine CHCOEF
NANG	code to indicate that third column of the array to be interpolated is a distance; called in subroutine CHCOEF
NCOURS	number of observations in COURSE array; read from file SIMDTA by subroutine INDAT
NCURNT	number of observations in XCURNT array; read from file SIMDTA by subroutine INDAT
NEDGE	number of observations in EDGE array; read from file SIMDTA by subroutine INDAT
NPT	number of observations in the array to be interpolated; used in subroutine INTER

NRUN1	first two alphanumeric characters giving the run number; read from file SIMDTA and written on file OUT384 by subroutine INDAT
NRUN2	last two alphanumeric characters giving the run number; read from file SIMDTA and written on file OUT384 by subroutine INDAT
NUM	dummy variable to see if angle is greater than $\pm$ 360°; used in subroutine INTER
NVAR	number of variables in CSTVAR array (7) to be updated; used in subroutine SOLVE
NWIND	number of observations in XWIND array; read from file SIMDTA by subroutine INDAT
NWPB	number of observations in XYPB and NYPB arrays; read from ADSIM by subroutine INDAT
NXALP	number of observations in XALPHA array; read from file ADSIM by subroutine INDAT
NYPB(7)	number of observations in the last of three dimensions in XYOP array, XYOP(I,J,NYPB(1)); read from file ADSIM by subroutine INDAT
PI	constant value π (3.14159); initialized in MANVER
PN	intermediate variable in calculation of PSI; calculated in subroutine SSCON
PSI	difference between ship heading and course heading (radians); calculated in subroutine SSCON and CALDER
PSID	first derivative of PSI with respect to time; calculated in subroutine SSCON and CALDER
PSIDD	second serivative of PSI with respect to time; calculated in subroutine SSCON and CALDER
PSISS	PSI angle for steady state; calculated in subroutine SSCON
RADCON	conversion factor from degrees to radians; initialized in MANVER
RHOA	mass density of air; read from file SIMDTA by subroutine INDAT

RHOW	mass density of water; read from file SIMDTA by subroutine INDAT
ROW	open water resistance; calculated in subroutine RESIST
RUDBAR	rudder angle for current TDATA time period; calculated in subroutine SSCON and RUDDER
RUDMAX	maximum allowable rudder angle, listed in degrees and converted to radians; read from file SIMDTA by subroutine INDAT
RWIND	wind resistance; calculated by subroutine WIND
SAMP(30)	array for printing results; printed in MANVER and subroutine PRT

1) TIME	8) YODD	15) location of max. excursion
2) XO	9) HSHIP	16) max. excursion at point
3) XOD	10) PSID/RADCON	17) total max. excursion
4) XODD	11) PSIDD	18) initial max. excursion
5) YOI	12) XRUDDR/RADCON	19) PSI
6) YO	13) VCURNT	
7) YOD	14) HCURNT	

SLOPE	slope of the line between the two calculated points to see if estimates are approaching desired line; calculated in subroutine SSCON
STVR(15)	dummy array to transfer variables between subroutines; set in subroutine SOLVE and used in subroutine CALDER

1) XOD	4) YO	7) HSHIP
2) XO	5) YOI	8-15) Not Used
3) YOD	6) PSID	

STVRO(15)	increment from first estimate of derivatives times the time change; used in subroutine SOLVE
STVR1(15)	increment from second estimate of derivatives times the time change; used in subroutine SOLVE
STVR2(15)	increment from third estimate of derivatives times the time change; used in subroutine SOLVE
STVRD(15)	first derivatives of values in STVR array; calculated in subroutine CALDER and used in subroutine SOLVE
T	time at which derivatives are taken; calculated in subroutine SOLVE

TDATA	time (sec) at which variables are updated; calculated in MANVER
TH	angle between wind direction and course heading (radians); calculated in subroutine WIND
TIME	current time (sec) for which observations and calculations are made; initialized in MANVER and used in subroutine SOLVE
TPRT	time (sec) at which values for data are to be printed; calculated in MANVER
TSHIP	ship thrust (pounds); read from file SIMDTA by subroutine INDAT
TSTART	time (sec) at beginning of case; read from file SIMDTA by subroutine INDAT
VCURNT	current velocity (mph) interpolated from XCURNT array; used in subroutine CALDER
VEL	ship's true velocity relative to land; calculated in subroutine SSCON and CHCOEF
VSHIP	ship velocity read in miles per hour and converted to feet per second; read from file SIMDTA by subroutine INDAT
VWIND	wind velocity (mph) interpolated from XWIND array; used in subroutine WIND
VWX	wind velocity in x-direction; used in subroutine WIND
VWY	wind velocity in y-direction; used in subroutine WIND
WDEPTH	water depth in channel (feet); read from file SIMDTA by subroutine INDAT
WPB	ratio of channel width to beam; used in subroutine CHCOEF
XALPHA(11,3)	array of channel coefficients; read from file ADSIM by subroutine INDAT

XALPHA(N,1)	H/T; N=1, NXALP
XALPHA(N,2)	$\overline{x}$; N=1, NXALP
XALPHA(N,3)	α; N=1, NXALP

XBAR — center of action of $Y_o{}'$, interpolated from XALPHA array; used in subroutine INTER

XCD — current velocity in x-direction; used in subroutine SSCON and CALDER

XCURNT(10,3) — array of current data; read from file SIMDTA by subroutine INDAT

- XCURNT(N,1) — x_o location of Nth observation (feet); N=1, NCURNT
- XCURNT(N,2) — current velocity at Nth observation (mph); N=1, NCURNT
- XCURNT(N,3) — current direction at Nth observation (°T); N=1, NCURNT

XD — XOD/cosϕ; used in subroutine SSCON

XEND — x_o location (feet) of end of simulation; read from file ADSIM by subroutine INDAT

XG — distance from the origin to the center of gravity measured along the x axis; used in subroutine CALDER

XIZ — moment of inertia; used in subroutine CALDER

XLBP — ship length between perpendiculars (feet); read from file SIMDTA by subroutine INDAT

XLOA — ship overall length (feet); read from file SIMDTA by subroutine INDAT

XLWL — ship length at waterline (feet); read from file SIMDTA by subroutine INDAT

XM — ship mass; used in subroutine CALDER

XMCOEF(15) — array of ship coefficients used in estimating initial values of first and second order derivative variables; used in subroutine SSCON

XNPSDD — partial derivative of yaw moment with respect to yaw angle acceleration; used in subroutine SSCON and CALDER

XNPSID — partial derivative of yaw moment with respect to yaw angle velocity; used in subroutine SSCON and CALDER

XNRUD	partial derivative of yaw moment with respect to rudder angle; used in subroutines SSCON and CALDER
XNV	partial derivative of yaw moment with respect to ship velocity; used in subroutine SSCON and CALDER
XNVD	partial derivative of yaw moment with respect to ship acceleration; used in subroutine SSCON and CALDER
XNWIND	partial derivative of yaw moment with respect to wind; used in subroutine SSCON and CALDER
XNYO	partial derivative of yaw moment with respect to ship lateral movement; used in subroutine SSCON and CALDER
XO	distance down the channel the ship has traveled; used in subroutine SSCON and CALDER
XOD	first derivative of distance traveled with respect to time; used in subroutine SSCON and CALDER
XODD	second derivative of distance traveled with respect to time; used in subroutine SSCON and CALDER
XRUDDR	rudder angle required (radians); calculated in subroutine RUDDER
XRUDSS	rudder angle required in steady state; calculated in subroutine SSCON
XSIGN	dummy variable used in determining the correct sign of interpolated values; used in subroutine INTER
XSTART	x_o location (feet) at which simulation will start; read from file SIMDTA by subroutine INDAT
XUD	ship added mass; calculated in subroutine SSCON
XVEL	ship velocity in x-direction; used in subroutine SSCON and CALDER
XWIND(10,3)	array of wind speed and direction; read from file SIMDTA by subroutine INDAT

XWIND(N,1)	time of Nth observation (sec); N=1, NWIND
XWIND(N,2)	wind speed at Nth observation (mph); N=1, NWIND
XWIND(N,3)	wind direction at Nth observation (°T); N=1, NWIND

XWPB(7)	array of values of W/B; read from file ADSIM by subroutine INDAT
XYOP(7,11,12)	dimensionless side force, Y_o', as a function of W/B and y_o/B; used in subroutine CHCOEF
XYZ(10,3)	dummy array to be interpolated; used in subroutine INTER
XYZ1	known value corresponding to value in first column of XYZ array; used in subroutine INTER
XYZ2	value interpolated from second column of XYZ array; calculated in subroutine INTER
XYZ3	value interpolated from third column of XYZ array; calculated by subroutine INTER
Y1D	intermediate calculation in first calculated point in estimate of steady state YO; calculated in subroutine SSCON
Y1DF	intermediate calculation for first calculated point in estimate of steady state YO; calculated in subroutine SSCON
Y1N	intermediate calculation for first calculated point in estimate of steady state YO; calculated in subroutine SSCON
Y1NF	intermediate calculation for first calculated point in estimate of steady state YO; calculated in subroutine SSCON
Y2D	intermediate calculation for second calculated point in estimate of steady state YO; calculated in subroutine SSCON
Y2DF	intermediate calculation for second calculated point in estimate of steady state YO; calculated in subroutine SSCON
Y2N	intermediate calculation for second calculated point in estimate of steady state YO; calculated in subroutine SSCON
Y2NF	intermediate calculation in second calculated point in estimate of steady state YO; calculated in subroutine SSCON

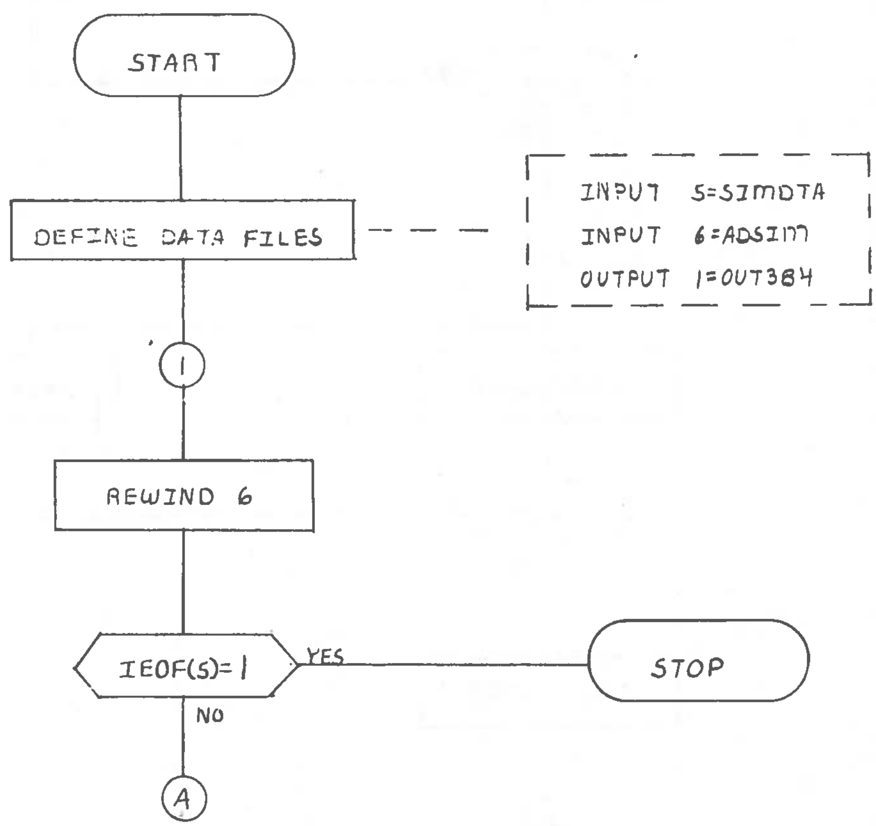
YCD	current velocity in y-direction; used in subroutine SSCON and CALDER
YCOURS	ships distance from centerline of channel (feet), interpolated from COURSE array; used in subroutine SSCON and CALDER
YCSS	ship course in steady state; calculated in SSCON
YO	distance of center of ship off center of channel (feet); used in subroutine SSCON and CALDER
YO1A	estimated value of first point to determine steady state YO; used in subroutine SSCON
YO1C	calculated value of first point to determine steady state YO; used in subroutine SSCON
YO2A	estimated value of second point to determine steady state YO; used in subroutine SSCON
YOD	first derivative of YO with respect to time; used in subroutine SSCON and CALDER
YODD	second derivative of YO with respect to time; used in subroutine SSCON and CALDER
YOFF	ships excursion off centerline (feet); calculated in MANVER
YOI	integral of $y_o - y_{course}$; calculated in subroutine SSCON and CALDER
YOP	Y_o', dimensionless side force; calculated in subroutine CHCOEF
YOP1	first value of YOP calculated at WPB less than actual; used in subroutine CHCOEF
YOP2	second value of YOP calculated at WPB greater than actual; used in subroutine CHCOEF
YOPP	intermediate interpolation value in calculating YOP; used in subroutine CHCOEF
YPSID	partial derivative of lateral force with respect to yaw angle velocity; used in subroutines SSCON and CALDER
YPSIDD	partial derivative of lateral force with respect to yaw angle acceleration; used in subroutines SSCON and CALDER

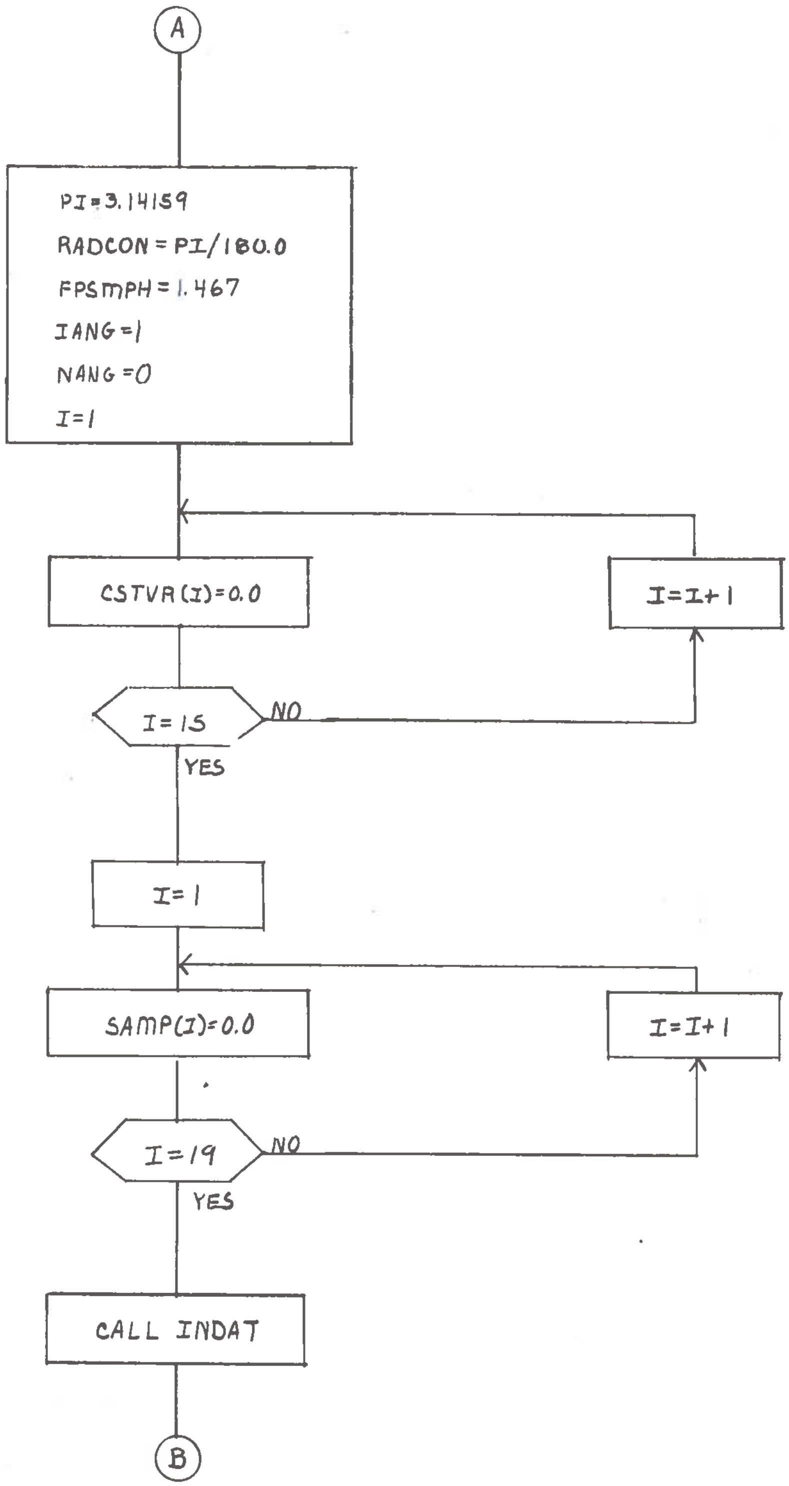
YRUD	partial derivative of lateral force with respect to rudder angle; used in subroutines SSCON and CALDER
YV	partial derivative of lateral force with respect to ship velocity; used in subroutines SSCON and CALDER
YVD	partial derivative of lateral force with respect to ship acceleration; used in subroutines SSCON and CALDER
YVEL	ship velocity in y-direction; used in subroutine SSCON and CALDER
YYO	partial derivative of lateral force with respect to ship lateral movement; used in subroutine SSCON and CALDER

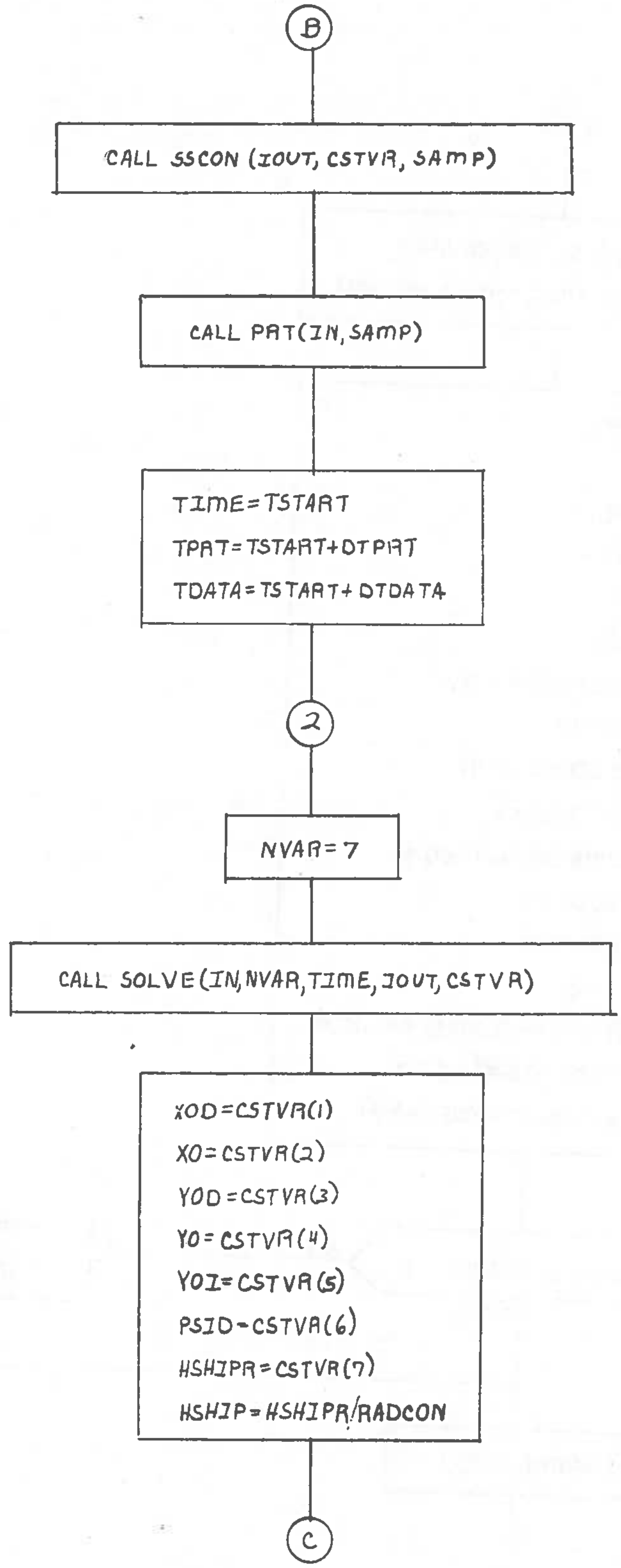
C.5 FLOW CHARTS

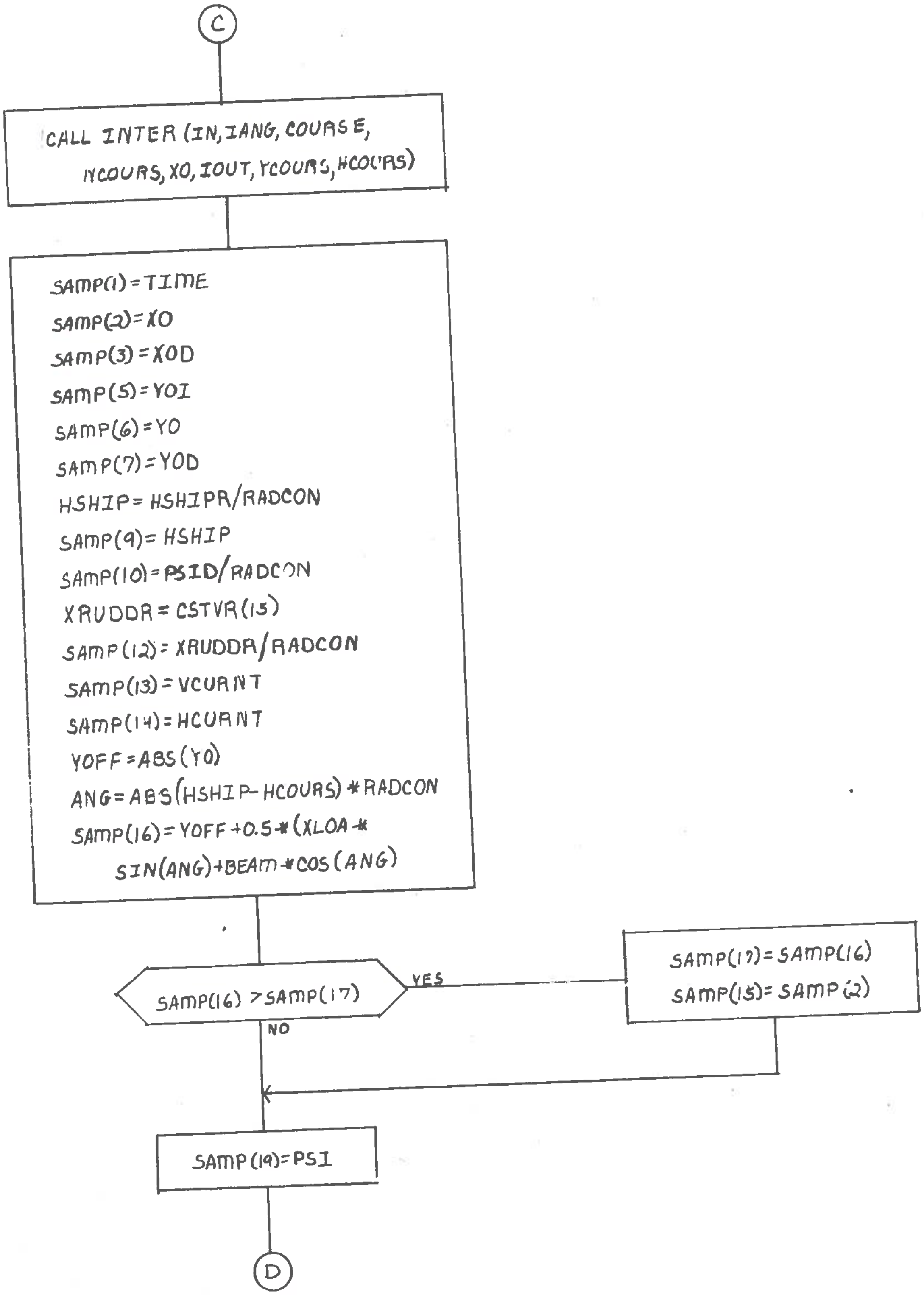
PROGRAM MANVER

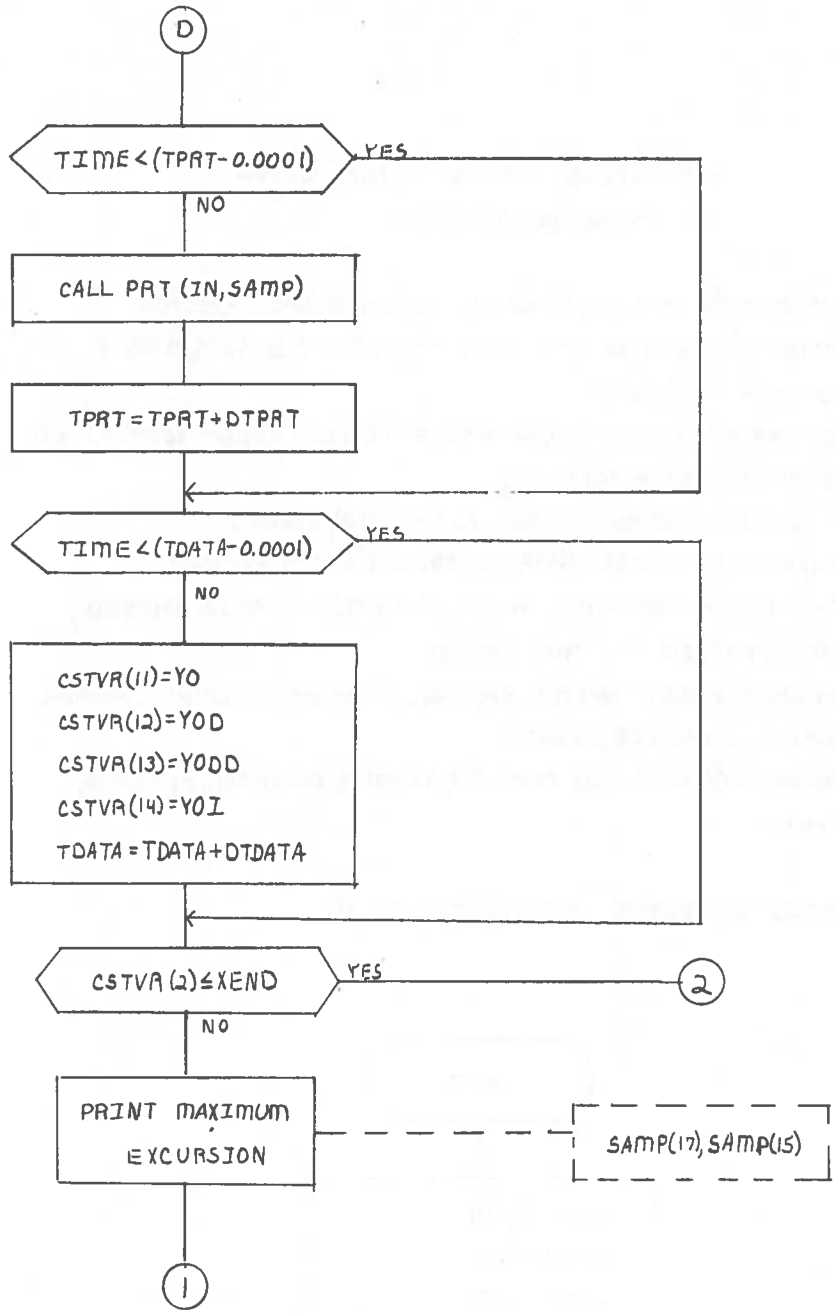
```
common/CONST/RHOA, RHOW, GRAV, PI, RADCON, DISPLC, FPSMPH
common/SIM/TSTART, XSTART, XEND, TPRT, TDATA, DTSIM, DTPRT,
    DTDATA, IANG, NANG
common/REACH/COURSE(10,3), NCOURS, EDGE(10,3), NEDGE, XCURNT(10,3),
    NCURNT, WDEPTH, YCSS, HCSS
common/WEATHR/VCURNT, HCURNT, XWIND(10,3), XWIND
common/CONTRL/CPOS(4), CHEAD(3), XRUDSS, PSISS, RUDBAR
common/COEFM/XMCOEF(15), XUD, YVD, YV, YPSIDD, YPSID, XNPSDD,
    XNVD, XNPSID, XNV, YRUD, XNRUD
common/SHIP/VSHIP, TSHIP, XLBP, XLWL, XLOA, BEAM, DRAFT, AREAP,
    AREAF, DISPL, XIZ, RUDMAX
common/COEFCH/XYOP(7,11,2), XWPB(7), NYPB(7), XALPHA(11,3), NWPB,
    NXALP

DIMENSION CSTVR(15), SAMP(19)
```



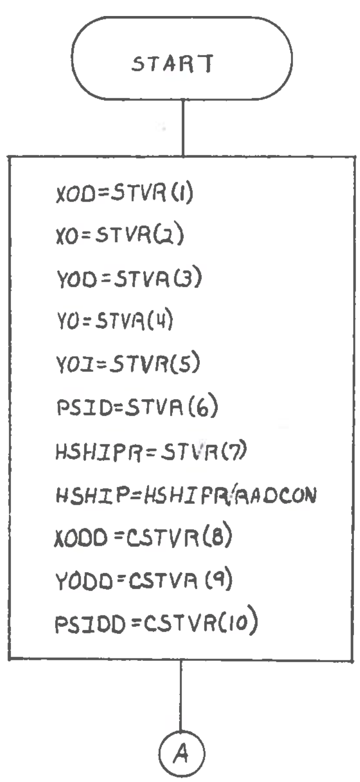

A
PI=3.14159
RADCON=PI/180.0
FPSMPH=1.467
IANG=1
NANG=0
I=1
CSTVR(I)=0.0
I=I+1
I=15
NO
YES
I=1
SAMP(I)=0.0
I=I+1
I=19
NO
YES
CALL INDAT
B

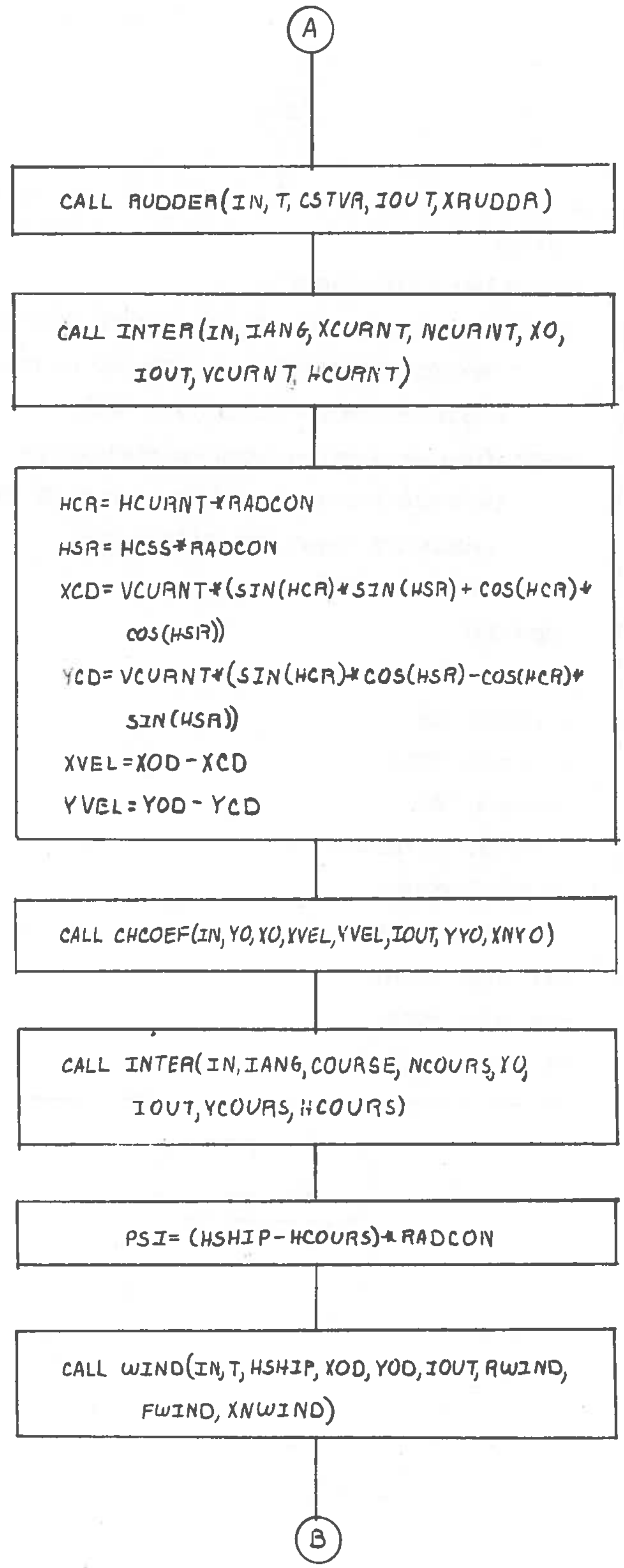

B
CALL SSCON (IOUT, CSTVR, SAMP)
CALL PAT(IN, SAMP)
TIME=TSTART
TPAT=TSTART+DTPAT
TDATA=TSTART+DTDATA
2
NVAR=7
CALL SOLVE(IN,NVAR,TIME,IOUT,CSTVR)
XOD=CSTVR(1)
XO=CSTVR(2)
YOD=CSTVR(3)
YO=CSTVR(4)
YOI=CSTVR(5)
PSID=CSTVR(6)
HSHIPR=CSTVR(7)
HSHIP=HSHIPR/RADCON
C


C
CALL INTER (IN, IANG, COURSE, NCOURS, XO, IOUT, YCOURS, HCOURS)
SAMP(1) = TIME
SAMP(2) = XO
SAMP(3) = XOD
SAMP(5) = YOI
SAMP(6) = YO
SAMP(7) = YOD
HSHIP = HSHIPR/RADCON
SAMP(9) = HSHIP
SAMP(10) = PSID/RADCON
XRUDDR = CSTVR(15)
SAMP(12) = XRUDDR/RADCON
SAMP(13) = VCURNT
SAMP(14) = HCURNT
YOFF = ABS(YO)
ANG = ABS(HSHIP-HCOURS) * RADCON
SAMP(16) = YOFF + 0.5 * (XLOA * SIN(ANG) + BEAM * COS(ANG)
SAMP(16) > SAMP(17)
YES
NO
SAMP(17) = SAMP(16)
SAMP(15) = SAMP(2)
SAMP(19) = PSI
D


D
TIME<(TPRT-0.0001)
YES
NO
CALL PRT (IN,SAMP)
TPRT=TPRT+DTPRT
TIME<(TDATA-0.0001)
YES
NO
CSTVR(11)=YO
CSTVR(12)=YOD
CSTVR(13)=YODD
CSTVR(14)=YOI
TDATA=TDATA+DTDATA
CSTVR(2)≤XEND
YES
2
NO
PRINT MAXIMUM EXCURSION
SAMP(17), SAMP(15)
1

```
SUBROUTINE CALDER(IN, NVAR, T, STVR,
      CSTVR, IOUT, STVRD)

COMMON/CONST/RHOA, RHOW, GRAV, PI, RADCON, DISPLC, FPSMPH
COMMON/SIM/TSTART, XSTART, XEND, TPRT, TDATA, DTSIM, DTPRT,
     DTDATA, IANG, NANG
COMMON/REACH/COURSE(10,3), NCOURS, EDGE(10,3), NEDGE, XCURNT(10,3),
     NCURNT, WDEPTH, YCSS, HCSS
COMMON/WEATHR/VCURNT, HCURNT, XWIND(10,3), NWIND
COMMON/CONTRL/CPOS(4), CHEAD(3), XRUDSS, PSISS, RUDBAR
COMMON/COEFM/XMCOEF(15), XUD, YVD, YV, YPSIDD, YPSID, XNPSDD,
     XNVD, XNPSID, XNV, YRUD, XNRUD
COMMON/SHIP/VSHIP, TSHIP, XLBP, XLWL, XLOA, BEAM, DRAFT, AREAP,
     AREAF, DISPL, XIZ, RUDMAX
COMMON/COEFCH/XYOP(7,11,2), XWPB(7), NYPB(7), XALPHA(11,3), NWPB,
     NXALP

DIMENSION STVR(15), STVRD(15), CSTVR(15)
```




A
CALL RUDDER(IN, T, CSTVR, IOUT, XRUDDR)
CALL INTER(IN, IANG, XCURNT, NCURNT, XO, IOUT, VCURNT, HCURNT)
HCR= HCURNT*RADCON
HSR= HCSS*RADCON
XCD= VCURNT*(SIN(HCR)*SIN(HSR)+COS(HCR)*COS(HSR))
YCD= VCURNT*(SIN(HCR)*COS(HSR)-COS(HCR)*SIN(HSR))
XVEL=XOD-XCD
YVEL=YOD-YCD
CALL CHCOEF(IN, YO, XO, XVEL, YVEL, IOUT, YYO, XNYO)
CALL INTER(IN, IANG, COURSE, NCOURS, XO, IOUT, YCOURS, HCOURS)
PSI= (HSHIP-HCOURS)*RADCON
CALL WIND(IN, T, HSHIP, XOD, YOD, IOUT, RWIND, FWIND, XNWIND)
B

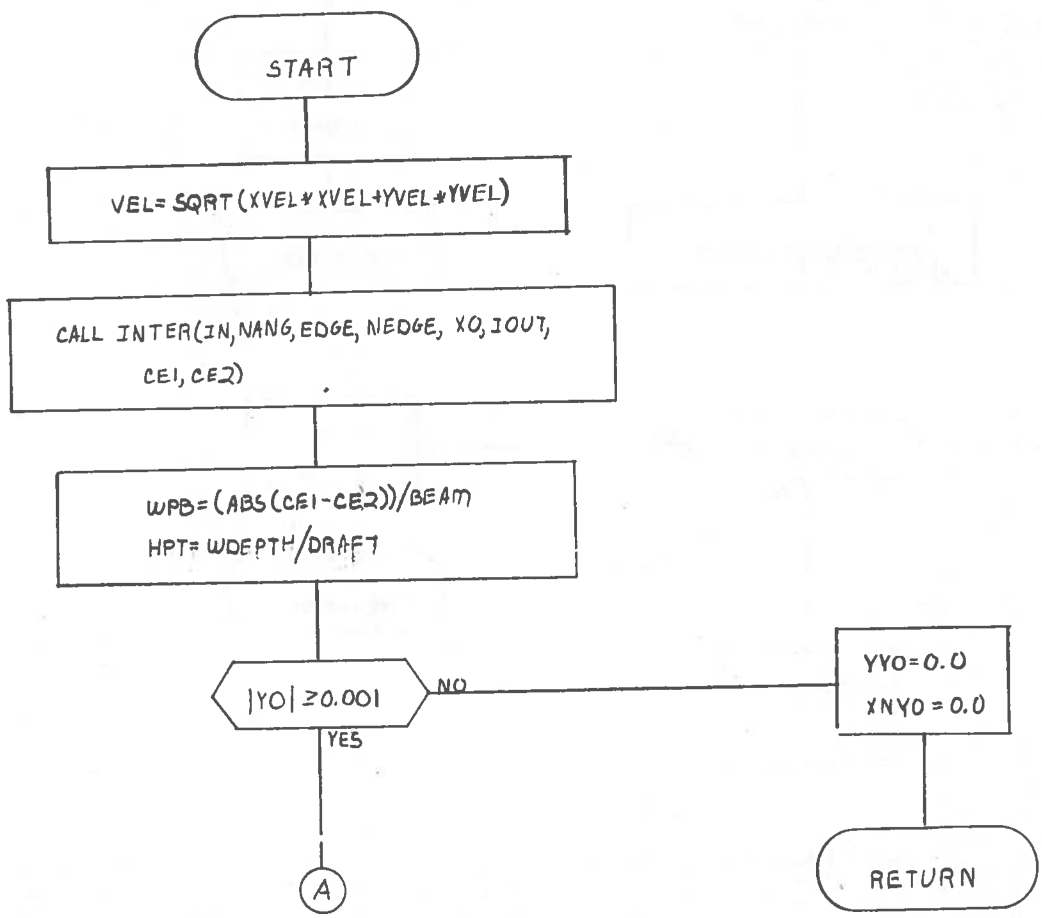
(B)

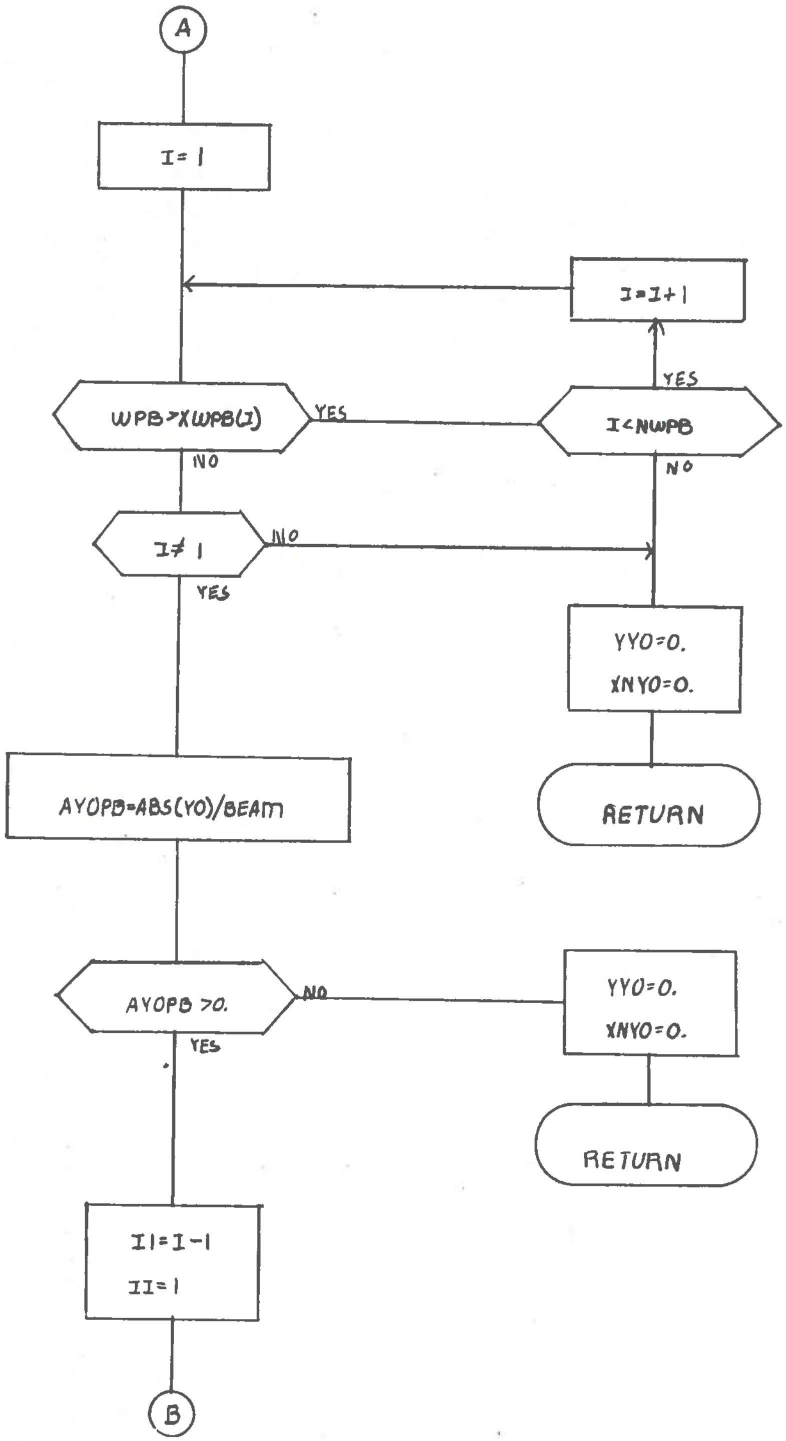
```
XG=0.0
XM=DISPLC*DISPL/GRAV
YODD=(FWIND+YRUD*XRUDDR+YYO*YO+YV*(YOD-YCD)
     -YV*(XOD-XCD)*PSI+(YPSID-YVD*XOD)*PSID+
     (YPSIDD-XM*XG)*PSIDD)/(XM-YVD)
PSIDD=(XNWIND+XNRUD*XRUDDR+XNYO*YO+XNV*
     (YOD-YCD)-XNV*(XOD-XCD)*PSI+(XNPSID-XNVD
     *XOD)*PSID+(XNVD-XM*XG)*YODD)/
     (XIZ-XNPSDD)
YODD=0.0
STVRD(1)=XODD
STVRD(2)=XOD
STVRD(3)=YODD
STVRD(4)=YOD
STVRD(5)=YO-YCOURS
STVRD(6)=PSIDD
STVRD(7)=FSID
CSTVR(8)=XODD
CSTVR(9)=YODD
CSTVR(10)=PSIDD
```

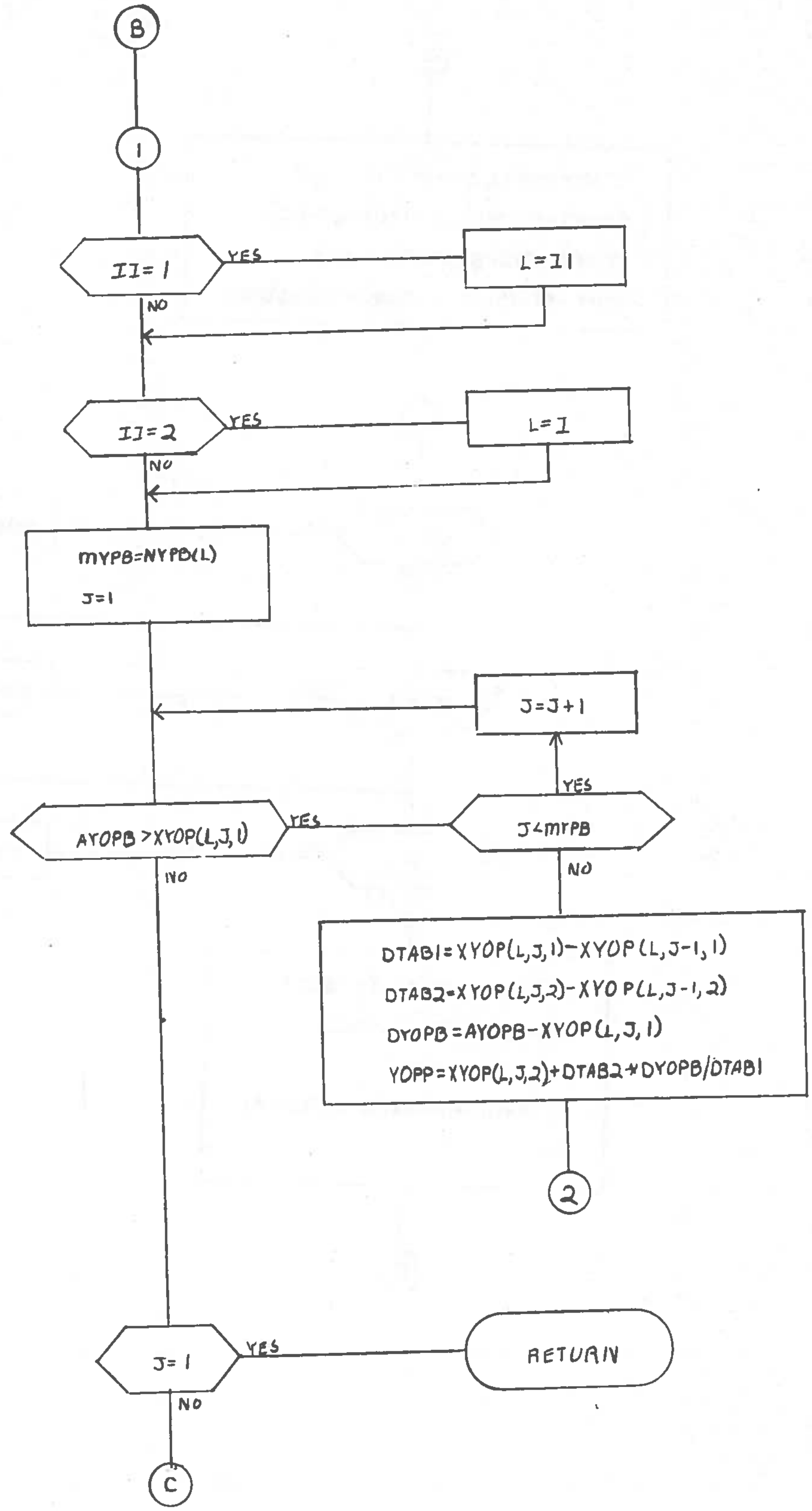
RETURN

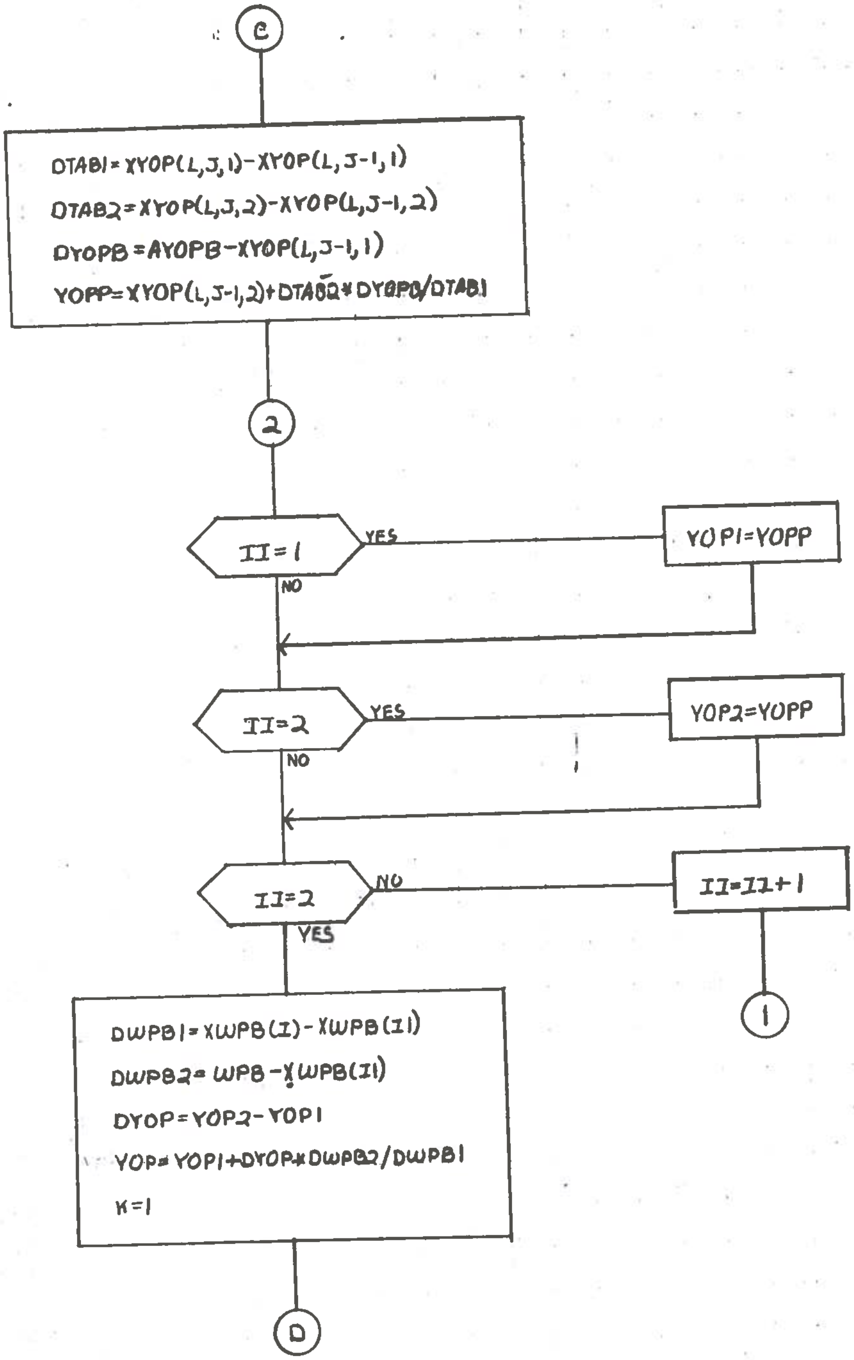
SUBROUTINE CHCOEF(IN,YO,XO,XVEL,YVEL,IOUT,YYO,XNYO)

```
common/CONST/RHOA, RHOW, GRAV, PI, RADCON, DISPLC, FPSMPH
common/SIM/TSTART, XSTART, XEND, TPRT, TDATA, DTSIM, DTPRT, DTDATA,
      IANG, NANG
common/REACH/COURSE(10,3), NCOURS, EDGE(10,3), NEDGE, XCURNT(10,3),
      NCURNT, WDEPTH, YCSS, HCSS
common/WEATHR/VCURNT, HCURNT, XWIND(10,3), NWIND
common/CONTRL/CPOS(4), CHEAD(3), XRUDSS, PSISS, RUDBAR
common/COEFM/XMCOEF(15), XUD, YVD, YV, YPSIDD, YPSID, XNPSDD, XNVD,
      XNPSID, XNV, YRUD, XNRUD
common/SHIP/VSHIP, TSHIP, XLBP, XLWL, XLOA, BEAM, DRAFT, AREAP,
      AREAF, DISPL, XIZ, RUDMAX
common/COEFCH/XYOP(7, 11, 2), XWPB(7), NYPB(7), YALPHA(11, 3),
      NWPB, NXALP
```

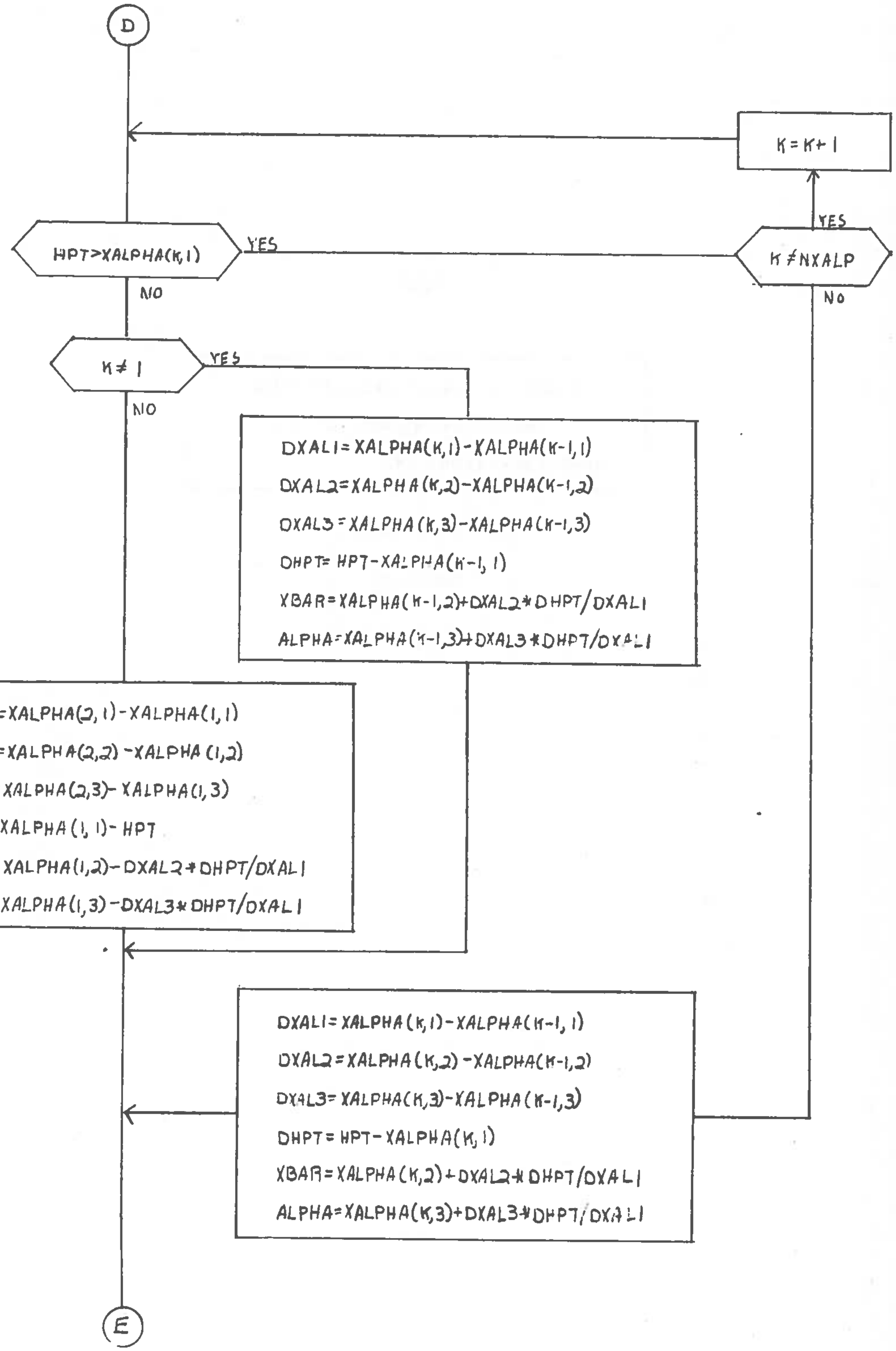


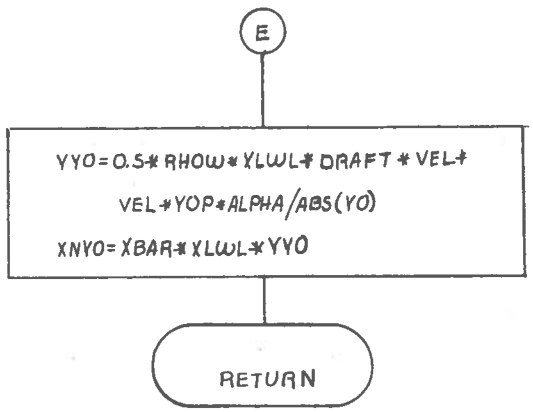

A
I=1
I=I+1
WPB>XWPB(I)
YES
NO
I<NWPB
YES
NO
I≠1
NO
YES
YYO=0.
XNYO=0.
AYOPB=ABS(YO)/BEAM
RETURN
AYOPB >0.
NO
YES
YYO=0.
XNYO=0.
RETURN
I1=I-1
II=1
B


B
1
II=1
YES
L=I1
NO
II=2
YES
L=I
NO
MYPB=NYPB(L)
J=1
J=J+1
YES
AYOPB>XYOP(L,J,1)
YES
J<MYPB
NO
NO
DTAB1=XYOP(L,J,1)-XYOP(L,J-1,1)
DTAB2=XYOP(L,J,2)-XYOP(L,J-1,2)
DYOPB=AYOPB-XYOP(L,J,1)
YOPP=XYOP(L,J,2)+DTAB2*DYOPB/DTAB1
2
J=1
YES
RETURN
NO
C



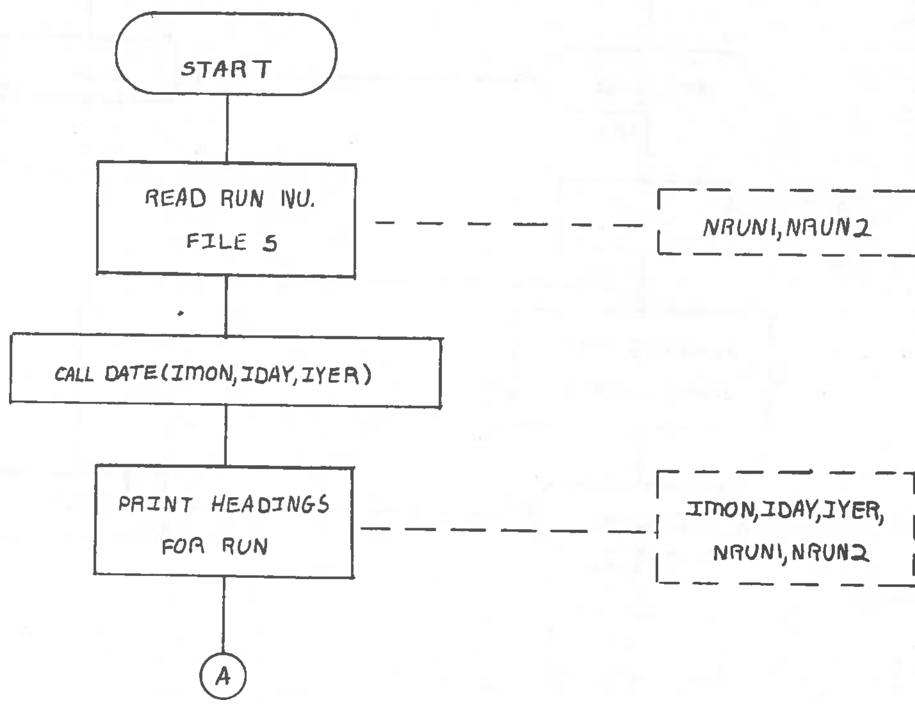
C
DTAB1=XYOP(L,J,1)-XYOP(L,J-1,1)
DTAB2=XYOP(L,J,2)-XYOP(L,J-1,2)
DYOPB=AYOPB-XYOP(L,J-1,1)
YOPP=XYOP(L,J-1,2)+DTAB2*DYOPB/DTAB1
2
II=1
YES
NO
YOP1=YOPP
II=2
YES
NO
YOP2=YOPP
II=2
NO
YES
II=II+1
1
DWPB1=XWPB(I)-XWPB(II)
DWPB2=WPB-XWPB(II)
DYOP=YOP2-YOP1
YOP=YOP1+DYOP*DWPB2/DWPB1
K=1
D

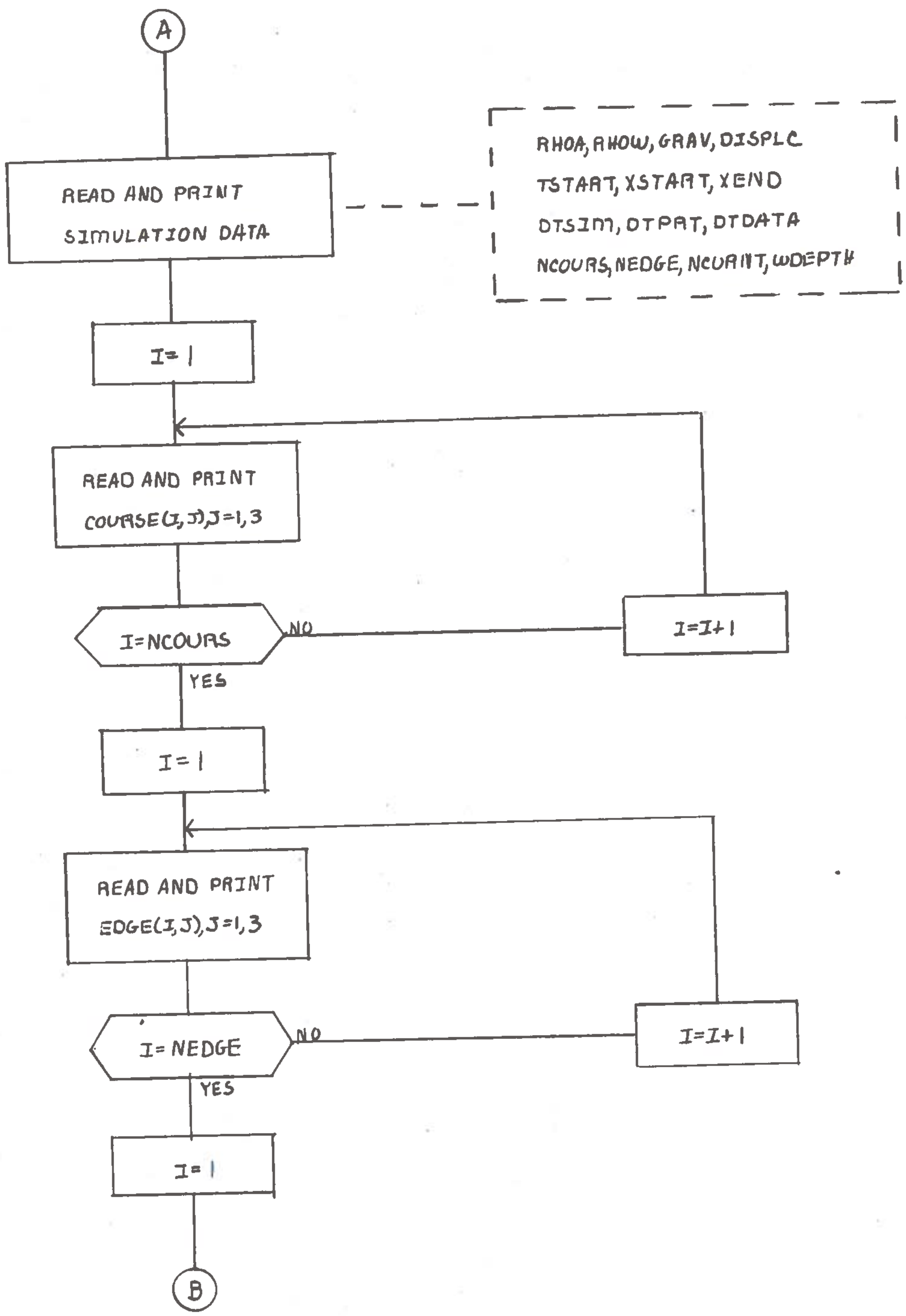

D
K = K+1
HPT>XALPHA(K,1)
YES
NO
K≠NXALP
YES
NO
K≠1
YES
NO
DXAL1=XALPHA(K,1)-XALPHA(K-1,1)
DXAL2=XALPHA(K,2)-XALPHA(K-1,2)
DXAL3=XALPHA(K,3)-XALPHA(K-1,3)
DHPT=HPT-XALPHA(K-1,1)
XBAR=XALPHA(K-1,2)+DXAL2*DHPT/DXAL1
ALPHA=XALPHA(K-1,3)+DXAL3*DHPT/DXAL1
DXAL1=XALPHA(2,1)-XALPHA(1,1)
DXAL2=XALPHA(2,2)-XALPHA(1,2)
DXAL3=XALPHA(2,3)-XALPHA(1,3)
DHPT=XALPHA(1,1)-HPT
XBAR=XALPHA(1,2)-DXAL2*DHPT/DXAL1
ALPHA=XALPHA(1,3)-DXAL3*DHPT/DXAL1
DXAL1=XALPHA(K,1)-XALPHA(K-1,1)
DXAL2=XALPHA(K,2)-XALPHA(K-1,2)
DXAL3=XALPHA(K,3)-XALPHA(K-1,3)
DHPT=HPT-XALPHA(K,1)
XBAR=XALPHA(K,2)+DXAL2*DHPT/DXAL1
ALPHA=XALPHA(K,3)+DXAL3*DHPT/DXAL1
E

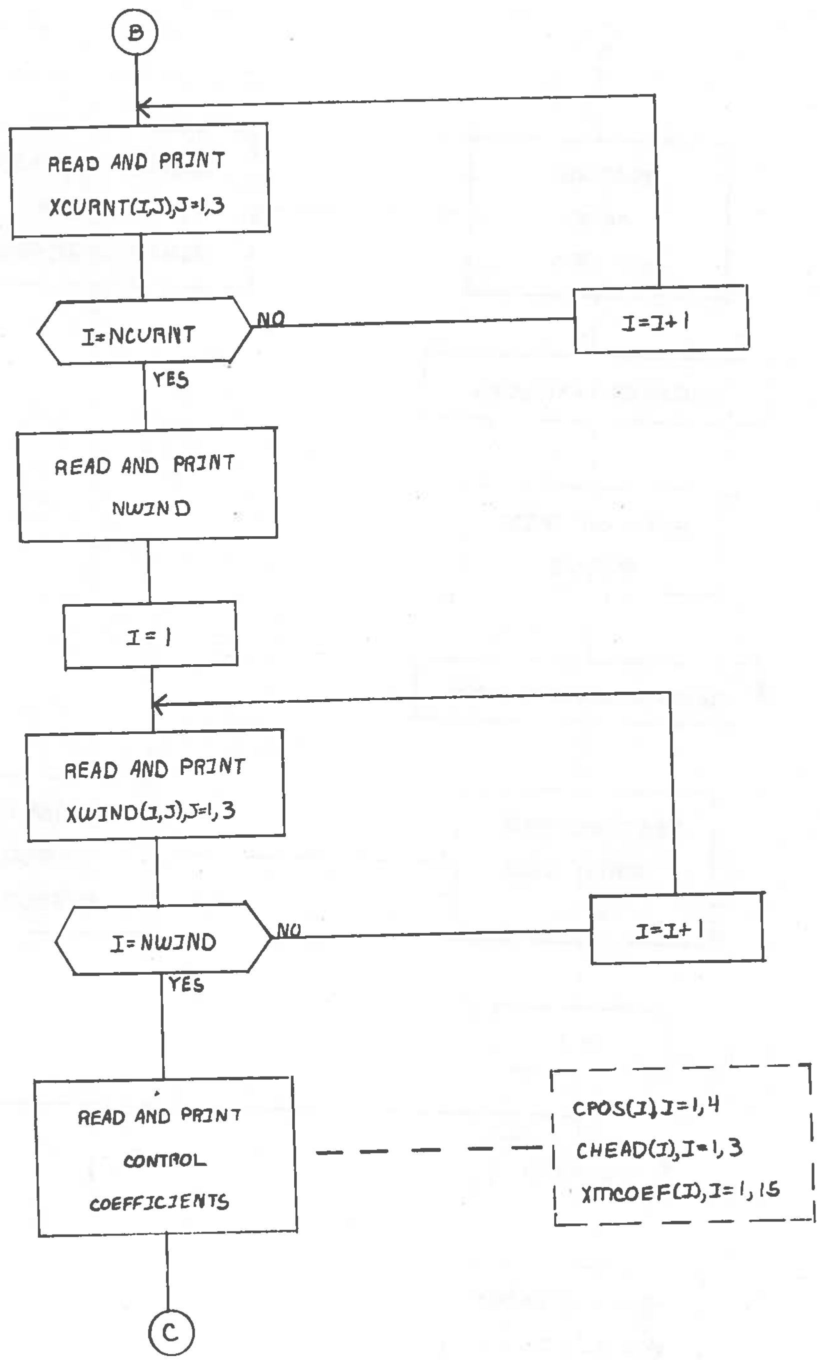

E
YYO=0.5*RHOW*XLWL*DRAFT*VEL*
VEL*YOP*ALPHA/ABS(YO)
XNYO=XBAR*XLWL*YYO
RETURN

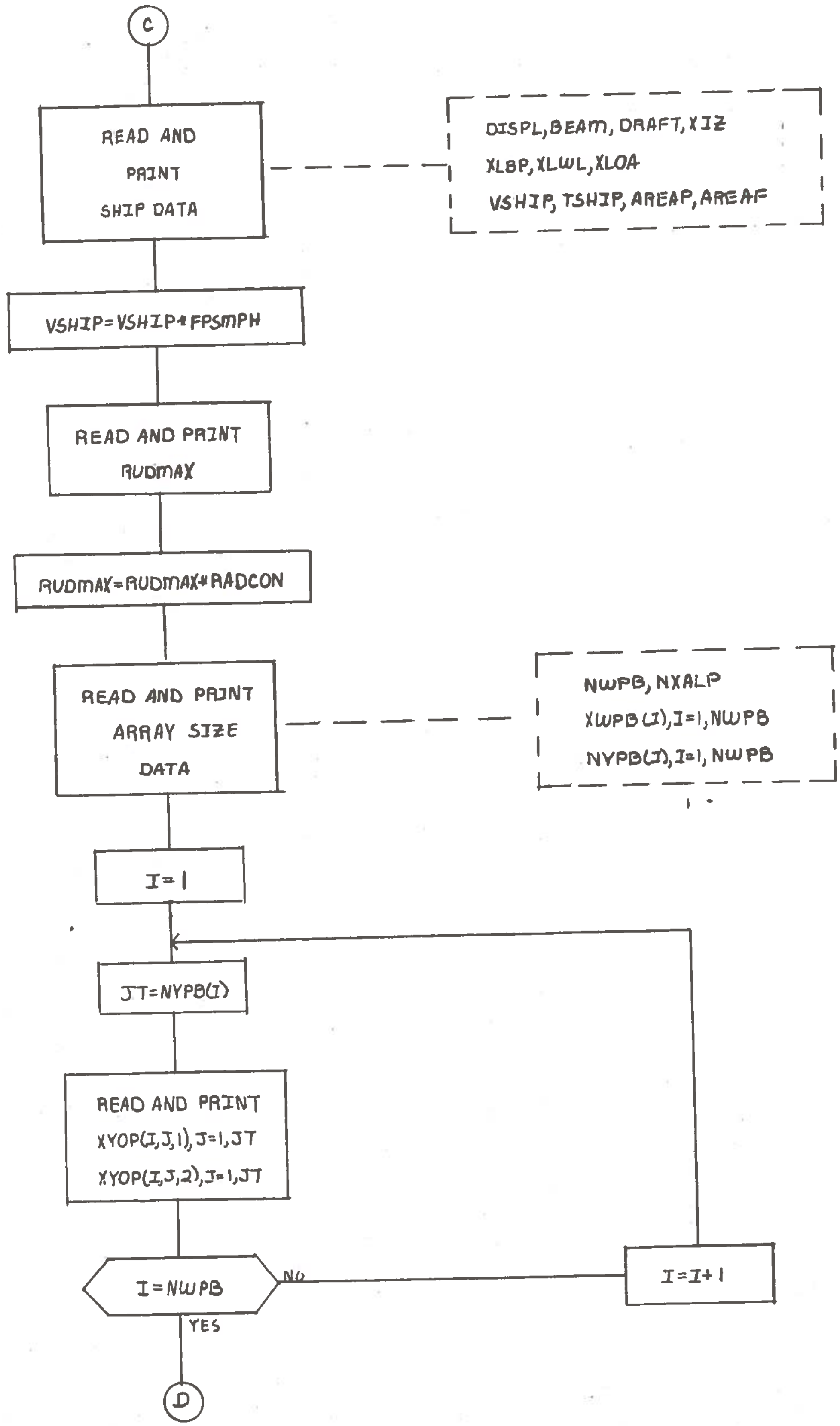
SUBROUTINE INDAT

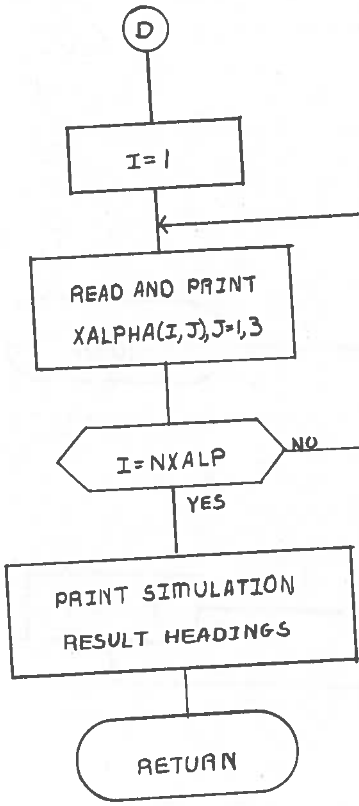
```
common/CONST/RHOA, RHOW, GRAV, PI, RADCON, DISPLC, FPSMPH
common/SIM/TSTART, XSTART, XEND, TPRT, TDATA, DTSIM, DTPRT,
      DTDATA, IANG, NANG
common/REACH/COURSE(10,3), NCOURS, EDGE(10,3), NEDGE, XCURNT(10,3),
      NCURNT, WDEPTH, YCSS, HCSS
common/WEATHR/VCURNT, HCURNT, XWIND(10,3), NWIND
common/CONTRL/CPOS(4), CHEAD(3), XRUDSS, PSISS, RUDEAR
common/COEFM/XMCOEF(15), XUD, YVD, YV, YPSIDD, YPSID, XNPSDD,
      XNVD, XNPSID, XNV, YRUD, XNRUD
common/SHIP/VSHIP, TSHIP, XLBP, XLWL, XLOA, BEAM, DRAFT, AREAP,
      AREAF, DISPL, XIZ, RUDMAX
common/COEFCH/XYOP(7,11,2), XWPB(7), NYPB(7), XALPHA(11,3),
      NWPB, NXALP

DIMENSION ITITLE(30)
```




A
READ AND PRINT SIMULATION DATA
RHOA, RHOW, GRAV, DISPLC
TSTART, XSTART, XEND
DTSIM, DTPRT, DTDATA
NCOURS, NEDGE, NCURNT, WDEPTH
I=1
READ AND PRINT COURSE(I,J),J=1,3
I=NCOURS
NO
YES
I=I+1
I=1
READ AND PRINT EDGE(I,J),J=1,3
I=NEDGE
NO
YES
I=I+1
I=1
B

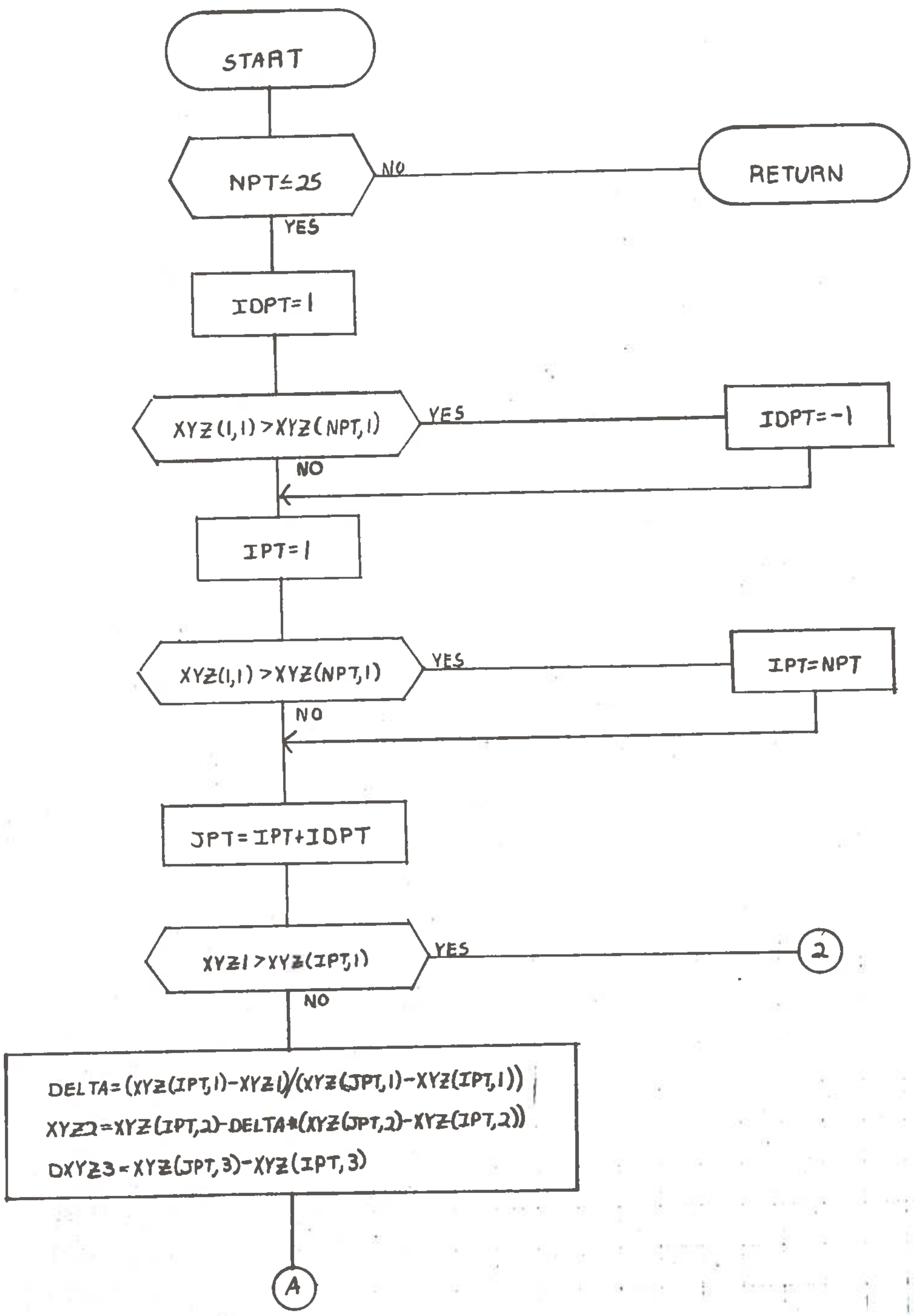

B
READ AND PRINT
XCURNT(I,J),J=1,3
I=NCURNT
NO
I=I+1
YES
READ AND PRINT
NWIND
I=1
READ AND PRINT
XWIND(I,J),J=1,3
I=NWIND
NO
I=I+1
YES
READ AND PRINT
CONTROL
COEFFICIENTS
CPOS(I),I=1,4
CHEAD(I),I=1,3
XMCOEF(I),I=1,15
C

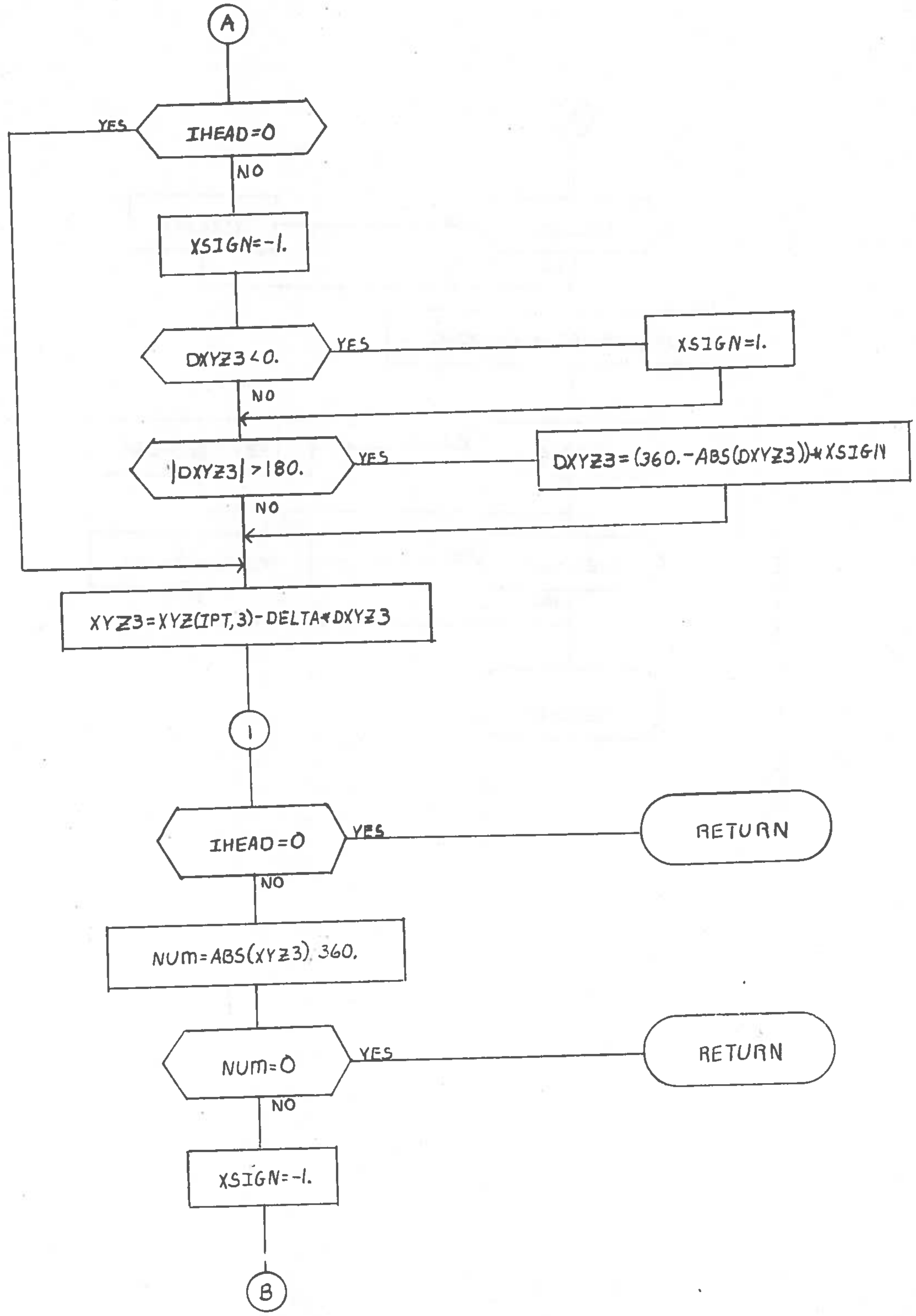

C
READ AND PRINT SHIP DATA
DISPL, BEAM, DRAFT, XIZ
XLBP, XLWL, XLOA
VSHIP, TSHIP, AREAP, AREAF
VSHIP=VSHIP*FPSMPH
READ AND PRINT RUDMAX
RUDMAX=RUDMAX*RADCON
READ AND PRINT ARRAY SIZE DATA
NWPB, NXALP
XWPB(I), I=1, NWPB
NYPB(I), I=1, NWPB
I=1
JT=NYPB(I)
READ AND PRINT
XYOP(I,J,1), J=1,JT
XYOP(I,J,2), J=1,JT
I=NWPB
NO
I=I+1
YES
D

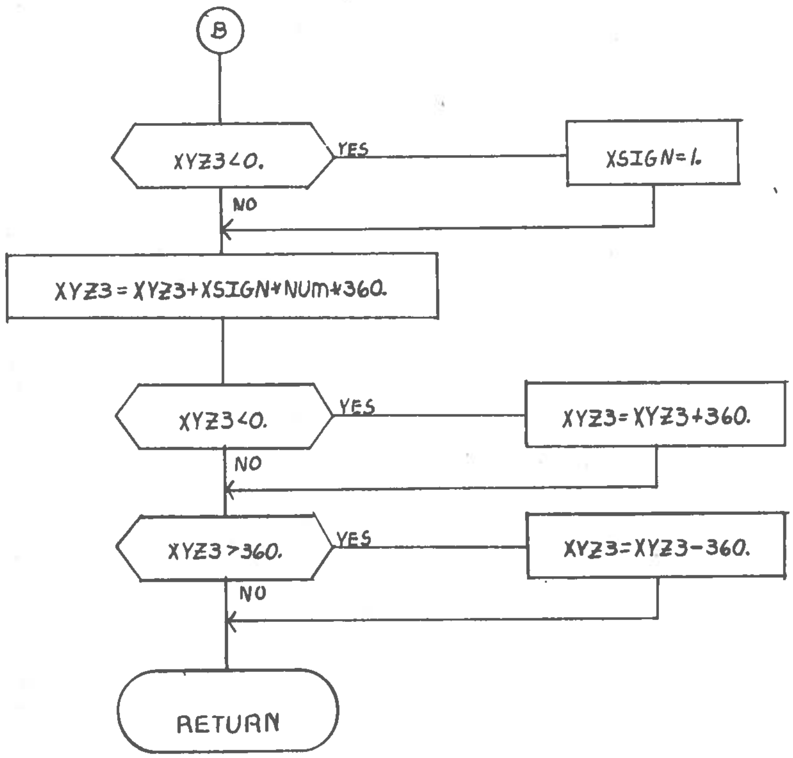

D
I=1
READ AND PRINT
XALPHA(I,J),J=1,3
I=NXALP
NO
YES
I=I+1
PRINT SIMULATION
RESULT HEADINGS
RETURN

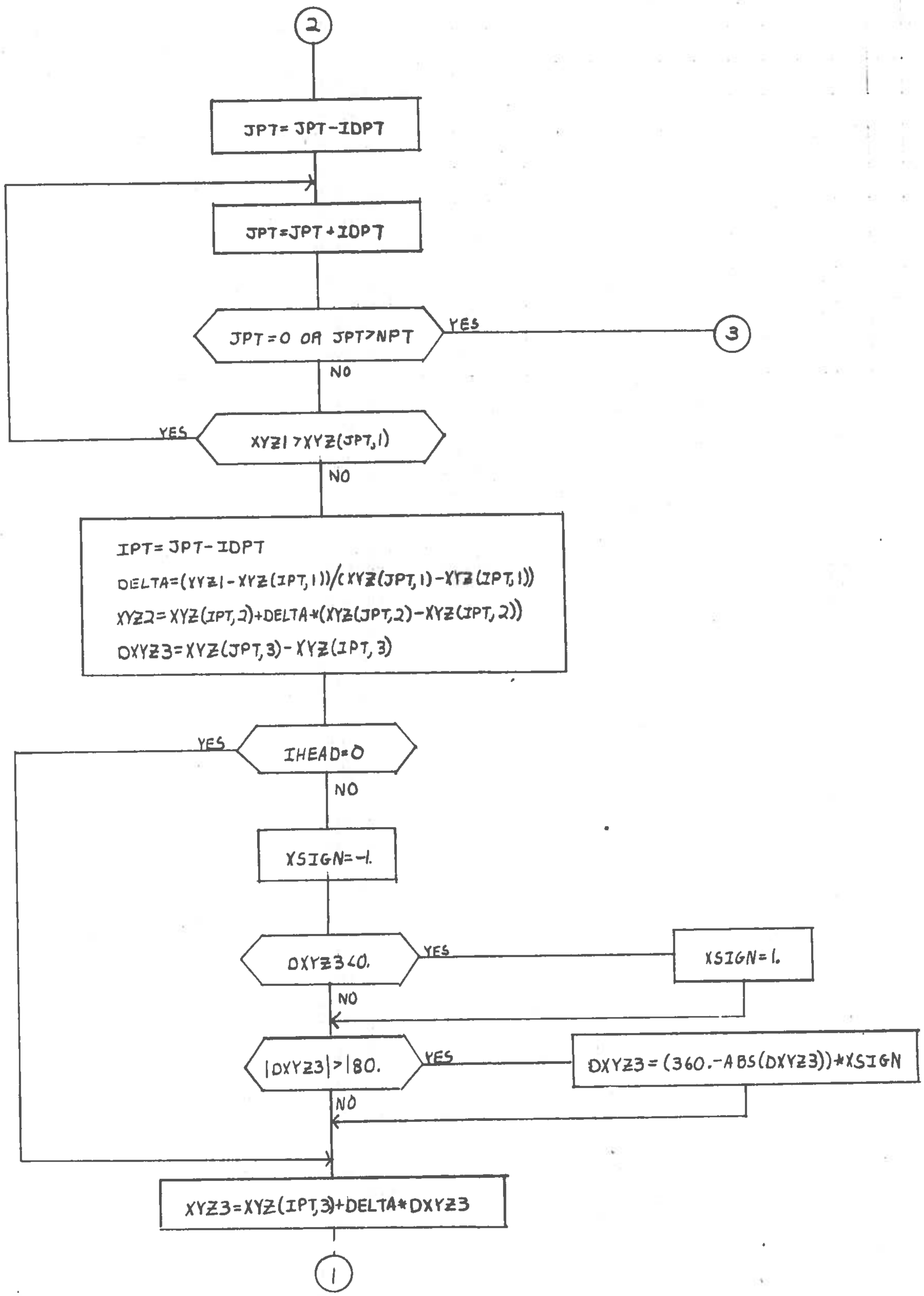
SUBROUTINE INTER(IN, IHEAD, XYZ, NPT, XYZ1, IOUT, XYZ2, XYZ3)

DIMENSION XYZ (10,3)

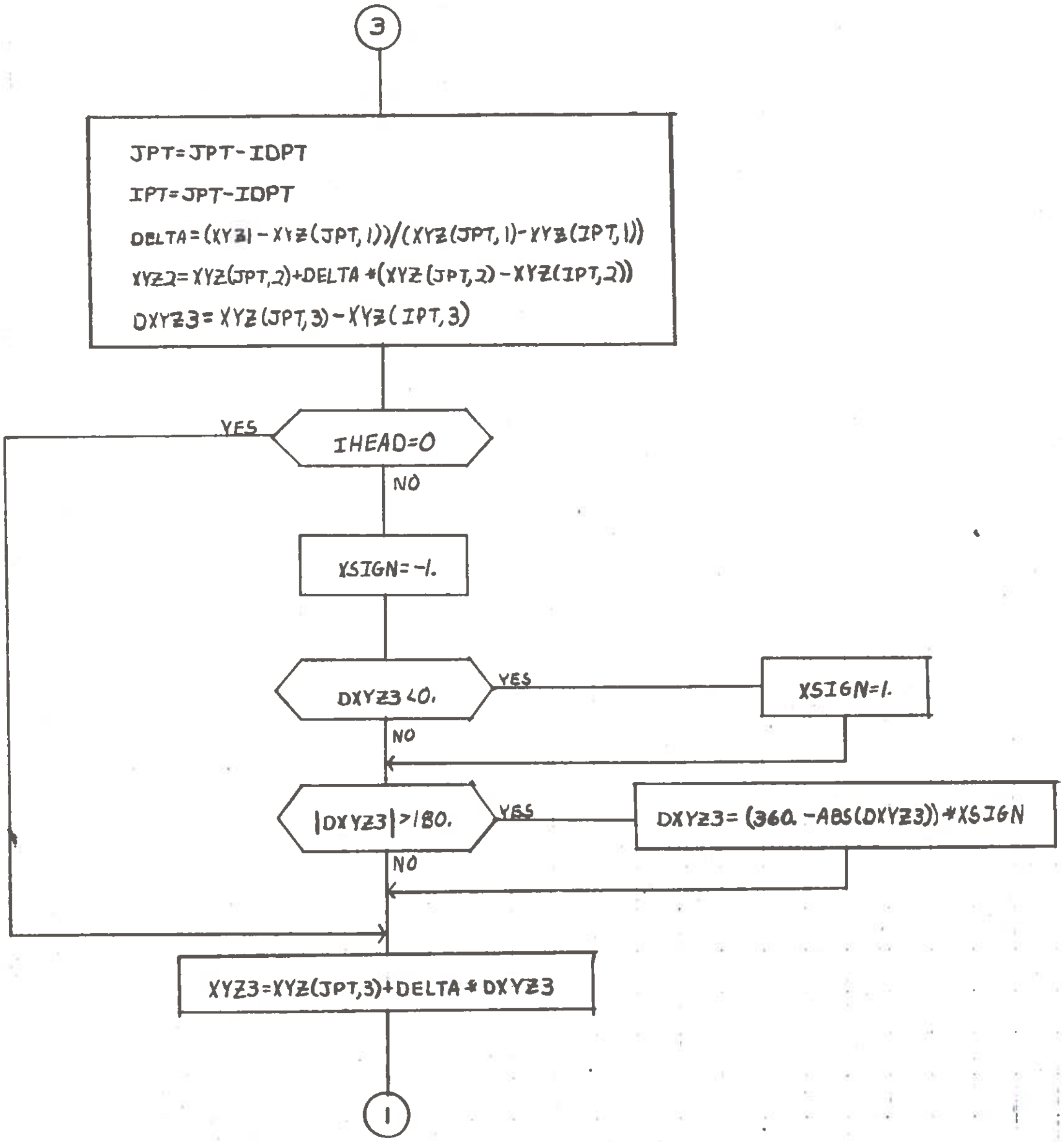



A
IHEAD=0
YES
NO
XSIGN=-1.
DXYZ3<0.
YES
XSIGN=1.
NO
|DXYZ3| >180.
YES
DXYZ3=(360.-ABS(DXYZ3))*XSIGN
NO
XYZ3=XYZ(IPT,3)-DELTA*DXYZ3
1
IHEAD=0
YES
RETURN
NO
NUM=ABS(XYZ3)/360.
NUM=0
YES
RETURN
NO
XSIGN=-1.
B


B
XYZ3<0.
YES
XSIGN=1.
NO
XYZ3=XYZ3+XSIGN*NUM*360.
XYZ3<0.
YES
XYZ3=XYZ3+360.
NO
XYZ3>360.
YES
XYZ3=XYZ3-360.
NO
RETURN

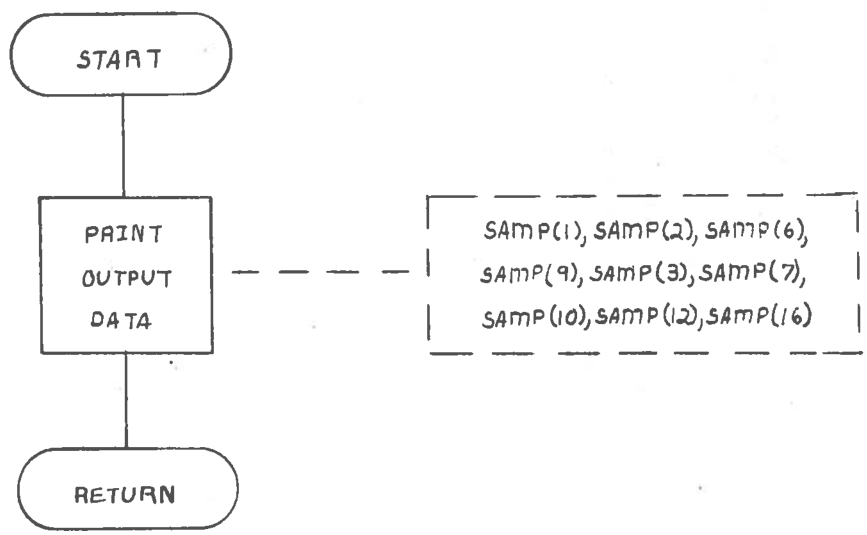


2
JPT= JPT-IDPT
JPT=JPT+IDPT
JPT=0 OR JPT>NPT
YES
3
NO
XYZ1>XYZ(JPT,1)
YES
NO
IPT=JPT-IDPT
DELTA=(XYZ1-XYZ(IPT,1))/(XYZ(JPT,1)-XYZ(IPT,1))
XYZ2=XYZ(IPT,2)+DELTA*(XYZ(JPT,2)-XYZ(IPT,2))
DXYZ3=XYZ(JPT,3)-XYZ(IPT,3)
IHEAD=0
YES
NO
XSIGN=-1.
DXYZ3<0.
YES
XSIGN=1.
NO
|DXYZ3|>180.
YES
DXYZ3=(360.-ABS(DXYZ3))*XSIGN
NO
XYZ3=XYZ(IPT,3)+DELTA*DXYZ3
1

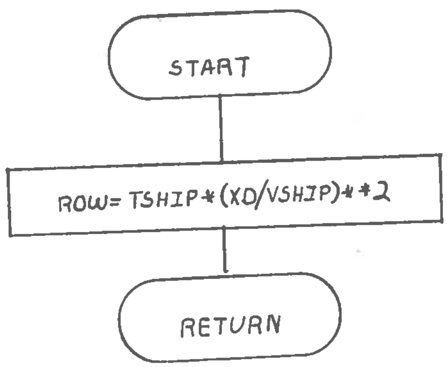

3
JPT=JPT-IDPT
IPT=JPT-IDPT
DELTA=(XYZ1-XYZ(JPT,1))/(XYZ(JPT,1)-XYZ(IPT,1))
XYZ2=XYZ(JPT,2)+DELTA*(XYZ(JPT,2)-XYZ(IPT,2))
DXYZ3=XYZ(JPT,3)-XYZ(IPT,3)
YES
IHEAD=0
NO
XSIGN=-1.
DXYZ3<0.
YES
XSIGN=1.
NO
|DXYZ3|>180.
YES
DXYZ3=(360.-ABS(DXYZ3))*XSIGN
NO
XYZ3=XYZ(JPT,3)+DELTA*DXYZ3
1

SUBROUTINE PRT(IN, SAMP)

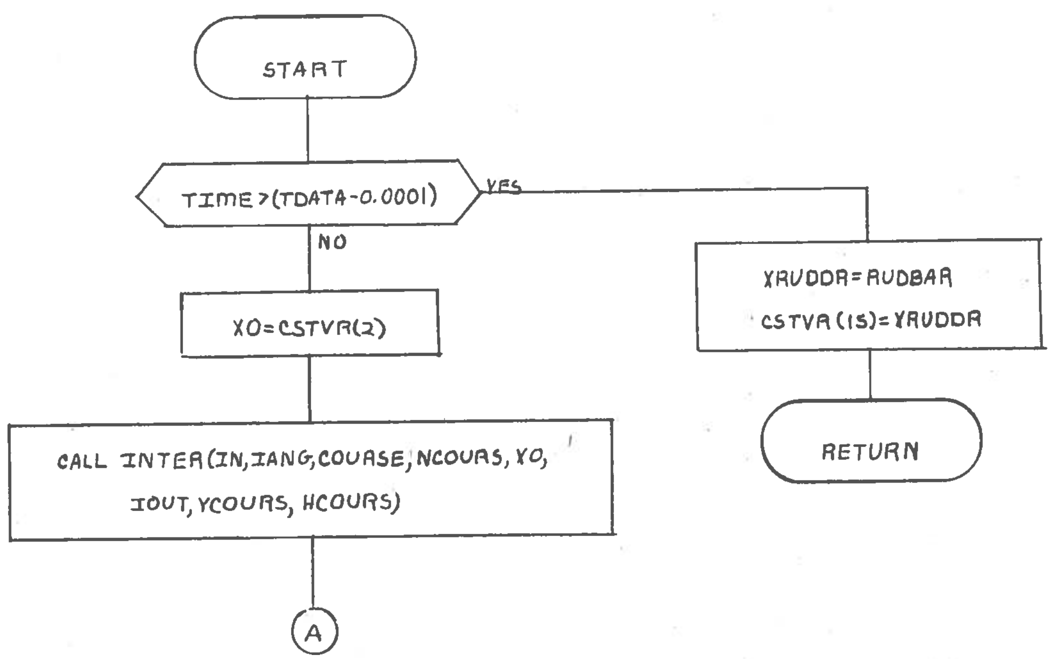
DIMENSION SAMP(19)

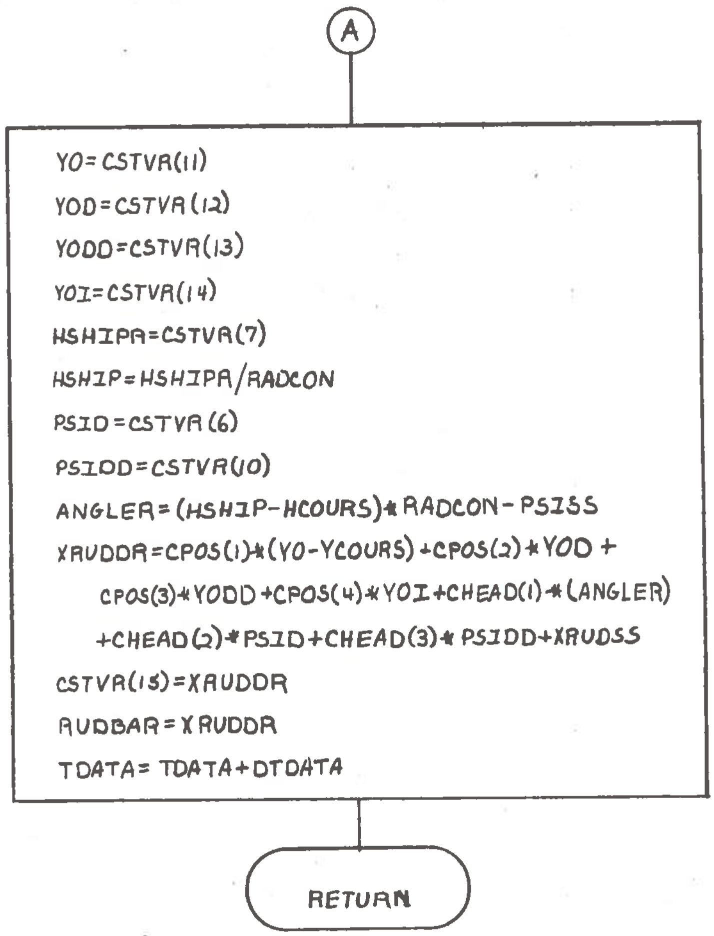


SUBROUTINE RESIST(IN, XD, TSHIP, VSHIP, IOUT, ROW)



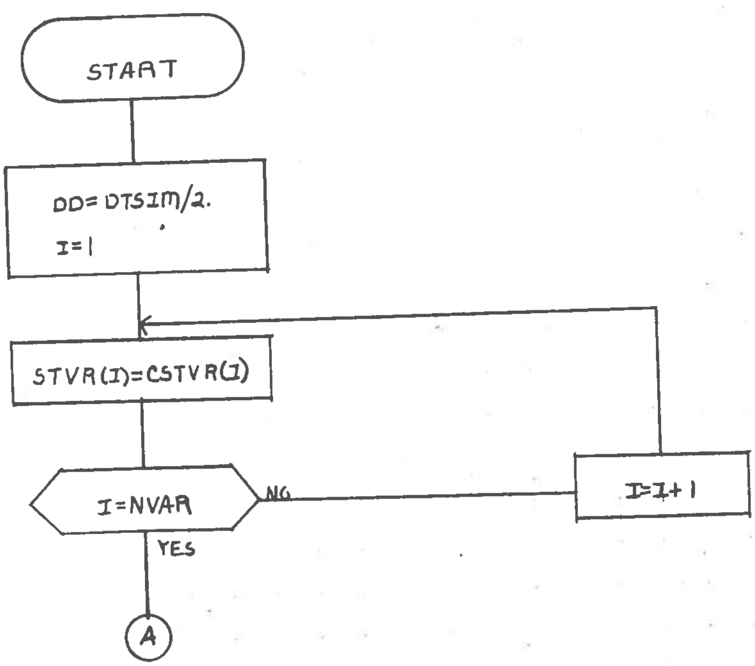
```
SUBROUTINE RUDDER(IN,TIME,CSTVR,IOUT,XRUDDR)

common/CONST/RHOA,RHOW,GRAV,PI,RADCON,DISPLC,FPSMPH
common/SIM/TSTART,XSTART,XEND,TPRT,TDATA,DTSIM,DTPRT,
     DTDATA,IANG,NANG
common/REACH/COURSE(10,3),NCOURS,EDGE(10,3),NEDGE,XCURNT(10,3),
     NCURNT,WDEPTH,YCSS,HCSS
common/WEATHR/VCURNT,HCURNT,XWIND(10,3),NWIND
common/CONTRL/CPOS(4),CHEAD(3),XRUDSS,PSISS,RUDBAR
common/COEFM/XMCOEF(15),XUD,YVD,YV,YPSIDD,YPSID,XNPSDD,
     XNVD,XNPSID,XNV,YRUD,XNRUD
common/SHIP/VSHIP,TSHIP,X_BP,X_WL,X_OA,BEAM,DRAFT,AREAP,
     AREAF,DISPL,XIZ,RUDMAX
common/COEFCH/XYOP(7,11,2),XWPB(7),NYPB(7),XALPHA(11,3),NWPB,
     NYALP

DIMENSION CSTVR(15)
```

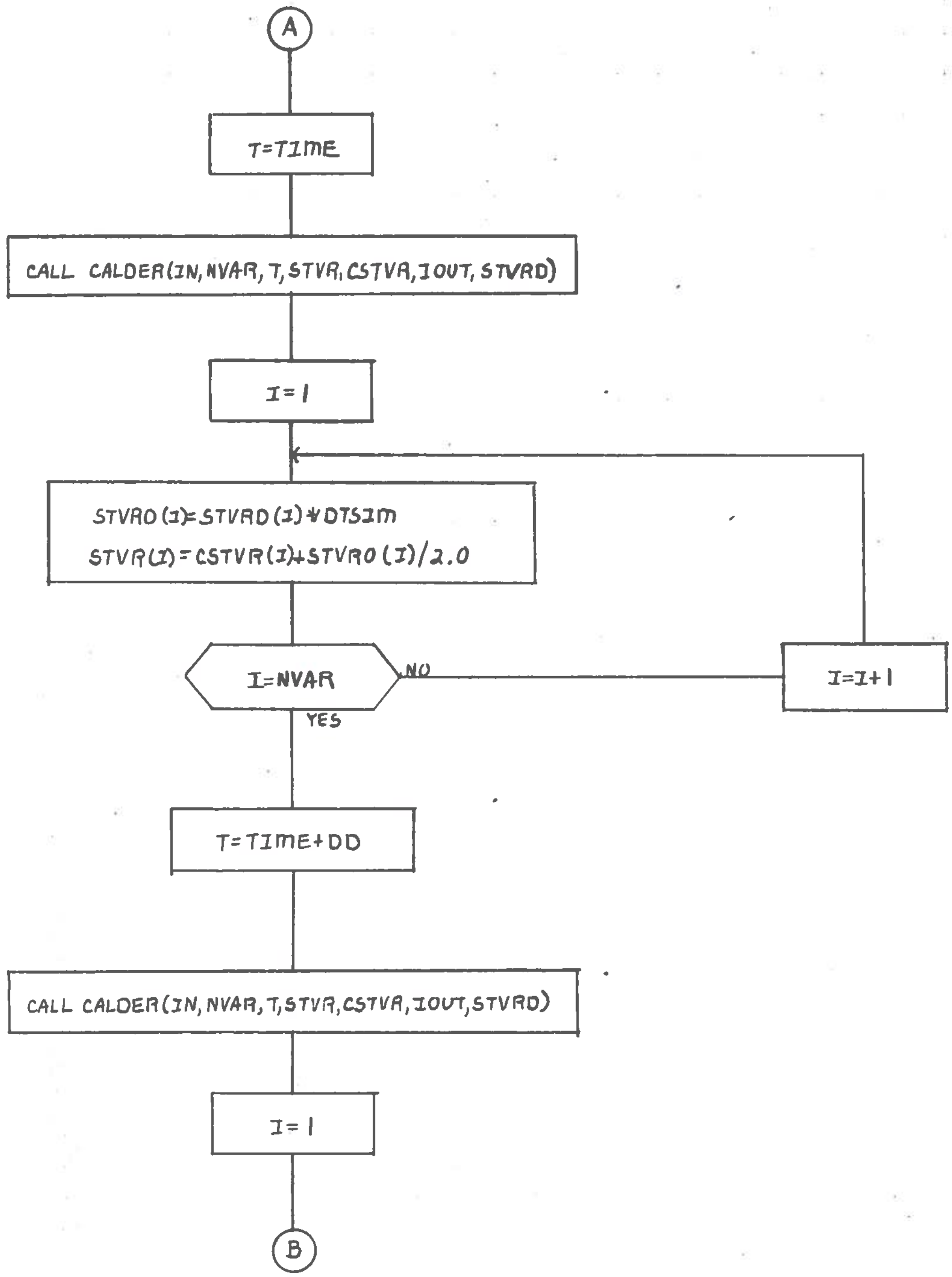


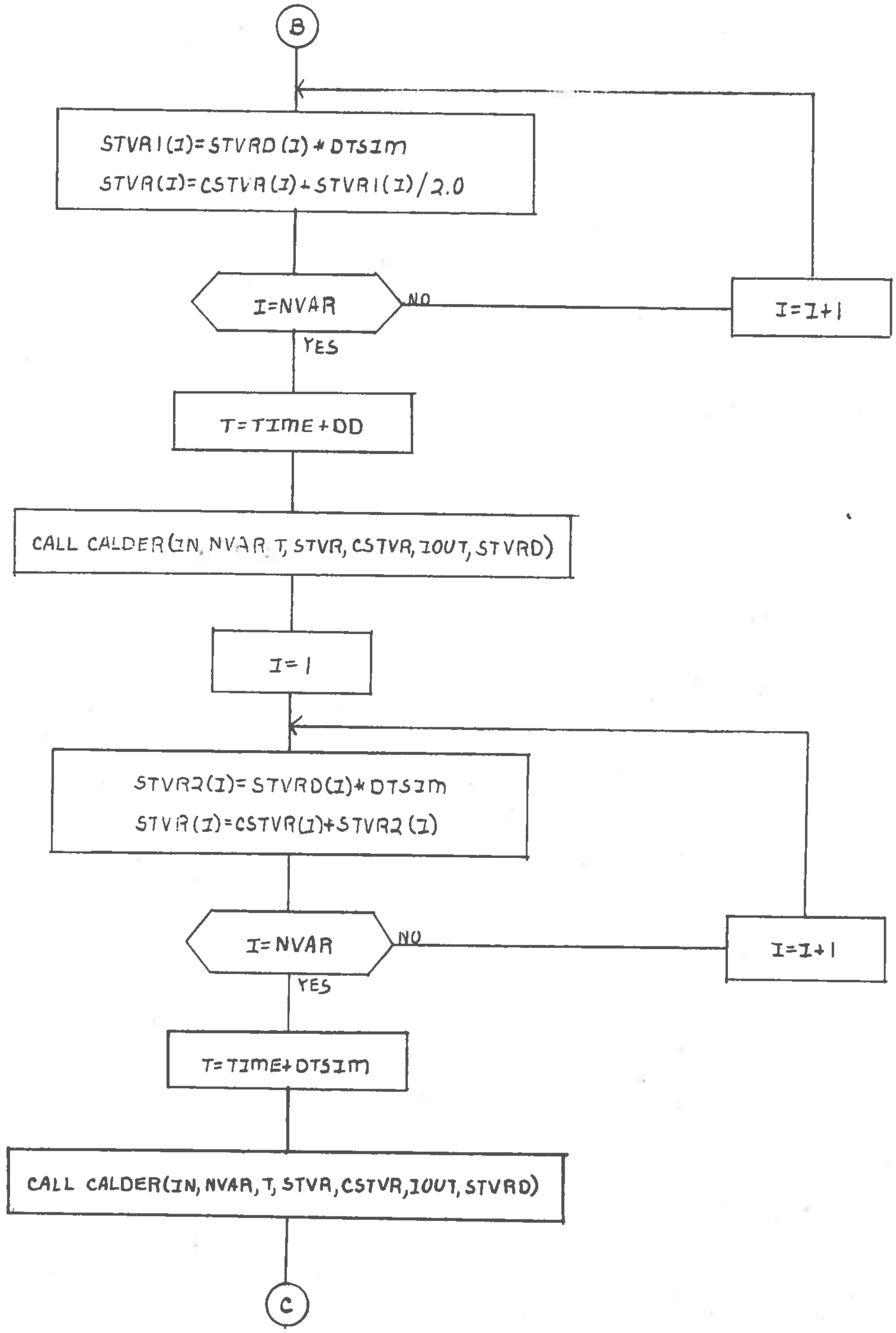

A
YO=CSTVR(11)
YOD=CSTVR(12)
YODD=CSTVR(13)
YOI=CSTVR(14)
HSHIPR=CSTVR(7)
HSHIP=HSHIPR/RADCON
PSID=CSTVR(6)
PSIDD=CSTVR(10)
ANGLER=(HSHIP-HCOURS)*RADCON-PSISS
XRUDDR=CPOS(1)*(YO-YCOURS)+CPOS(2)*YOD+
CPOS(3)*YODD+CPOS(4)*YOI+CHEAD(1)*(ANGLER)
+CHEAD(2)*PSID+CHEAD(3)*PSIDD+XRUDSS
CSTVR(15)=XRUDDR
RUDBAR=XRUDDR
TDATA=TDATA+DTDATA
RETURN

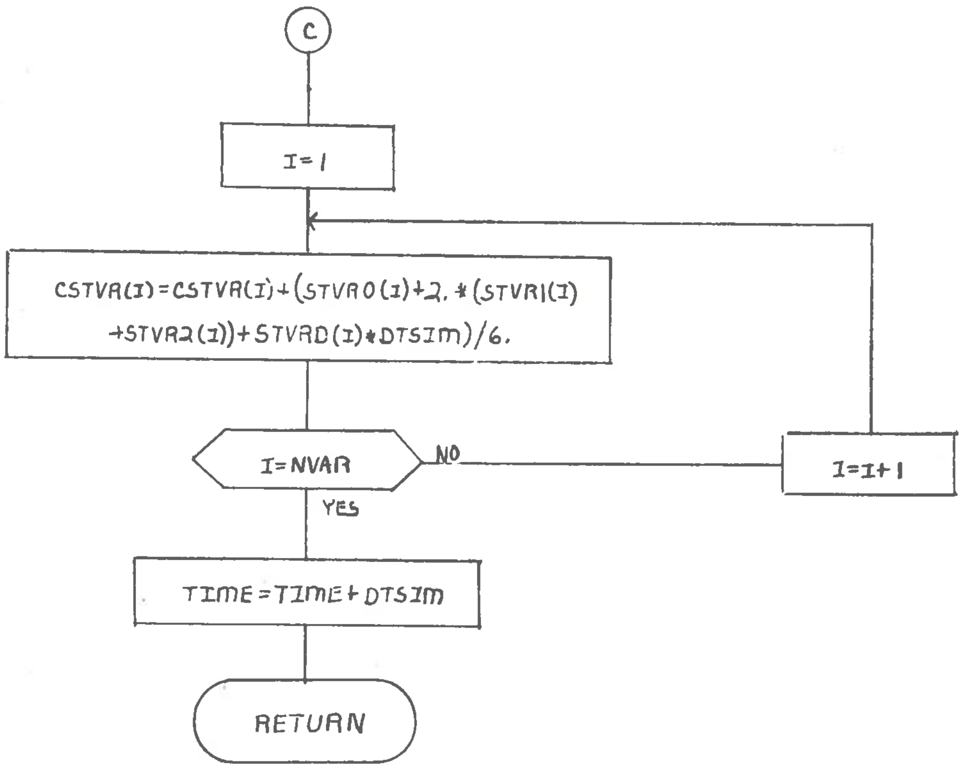
SUBROUTINE SOLVE(IN,NVAR,TIME,IOUT,CSTVR)

```
COMMON/CONST/RHOA,RHOW,GRAV,PI,RADCON,DISPLC,FPSMPH
COMMON/SIM/TSTART,XSTART,XEND,TPRT,TDATA,DTSIM,DTPRT,
     DTDATA,IANG,NANG
COMMON/REACH/COURSE(10,3),NCOURS,EDGE(10,3),NEDGE,XCURNT(10,3),
     NCURNT,WDEPTH,VCSS,HCSS
COMMON/WEATHR/VCURNT,HCURNT,XWIND(10,3),NWIND
COMMON/CONTRL/CPOS(4),CHEAD(3),XRUDSS,PSISS,RUDBAR
COMMON/COEFM/XMCOEF(15),XUD,YVD,YV,YPSIDD,YPSID,XNPSDD,
     XNVD,XNPSID,XNV,YRUD,XNRUD
COMMON/SHIP/VSHIP,TSHIP,XLBP,XLWL,XLOA,BEAM,DRAFT,AREAP,
     AREAF,DISPL,XIZ,RUDMAX
COMMON/CHCOEF/XYOP(7,11,2),XWPB(7),NYPB(7),XALPHA(11,3),
     NWPB,NXALP

DIMENSION STVR(15),CSTVR(15),STVRD(15),STVRO(15),STVR1(15),
     STVR2(15)
```

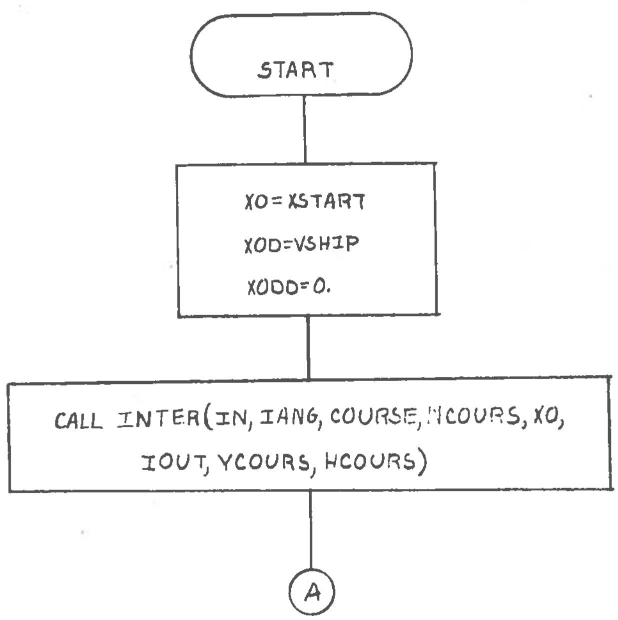


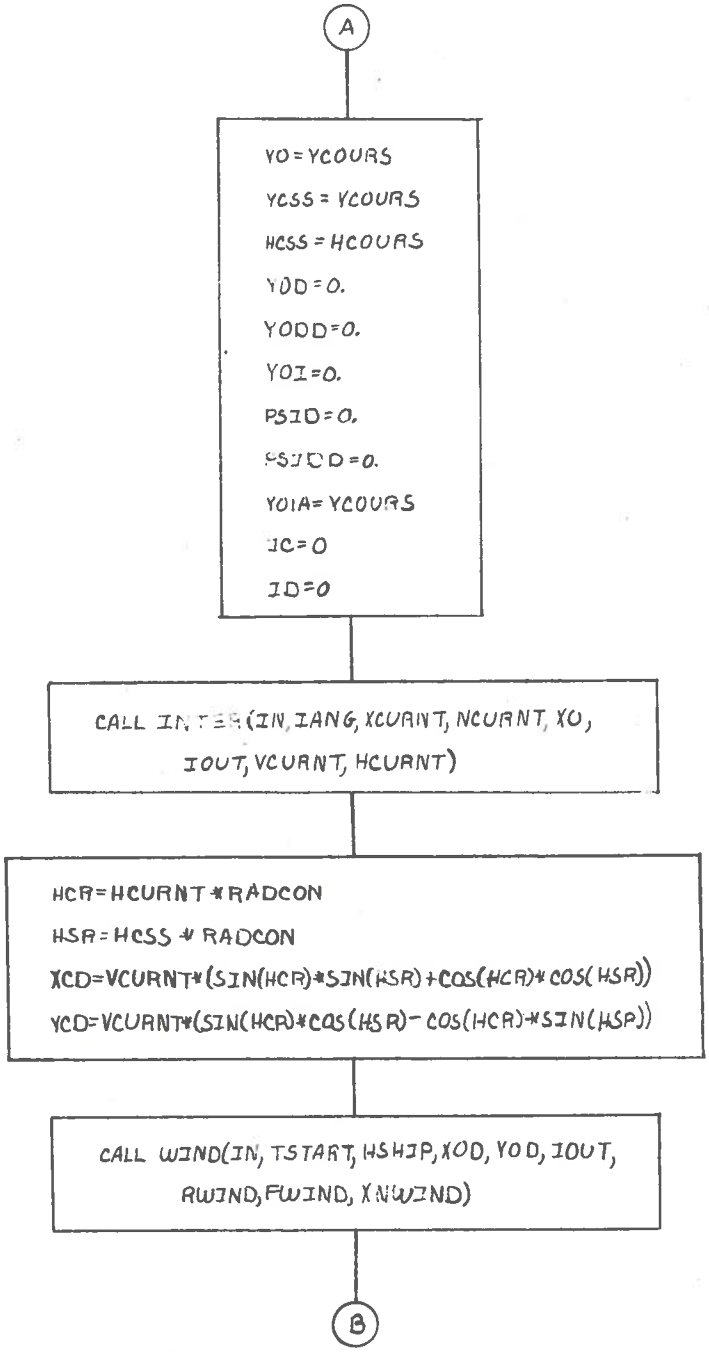

A
T=TIME
CALL CALDER(IN, NVAR, T, STVR, CSTVR, IOUT, STVRD)
I=1
STVRO(I)=STVRD(I)*DTSIM
STVR(I)=CSTVR(I)+STVRO(I)/2.0
I=NVAR
NO
YES
I=I+1
T=TIME+DD
CALL CALDER(IN, NVAR, T, STVR, CSTVR, IOUT, STVRD)
I=1
B


B
STVR1(I)=STVRD(I)*DTSIM
STVR(I)=CSTVR(I)+STVR1(I)/2.0
I=NVAR
NO
I=I+1
YES
T=TIME+DD
CALL CALDER(IN, NVAR, T, STVR, CSTVR, IOUT, STVRD)
I=1
STVR2(I)=STVRD(I)*DTSIM
STVR(I)=CSTVR(I)+STVR2(I)
I=NVAR
NO
I=I+1
YES
T=TIME+DTSIM
CALL CALDER(IN, NVAR, T, STVR, CSTVR, IOUT, STVRD)
C

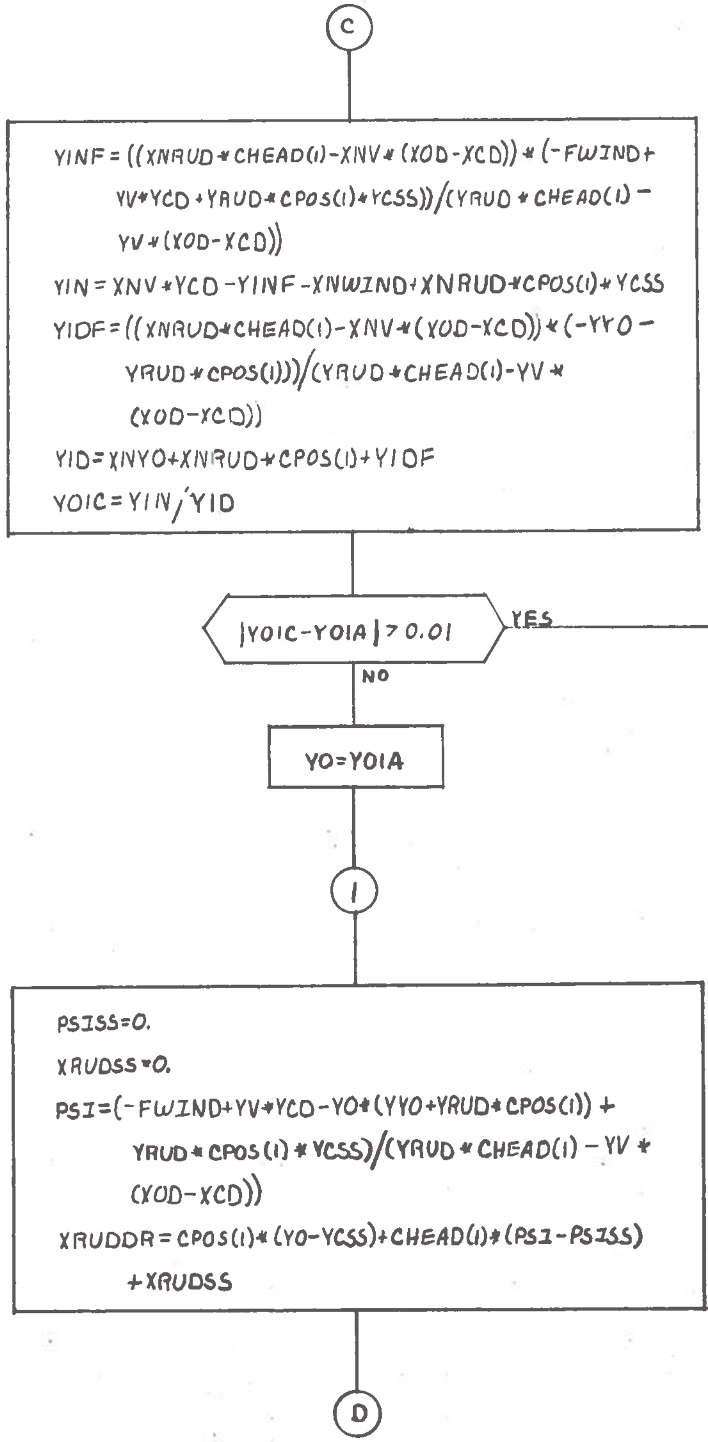

C
I=1
CSTVR(I)=CSTVR(I)+(STVR0(I)+2.*(STVR1(I)
+STVR2(I))+STVRD(I)*DTSIM)/6.
I=NVAR
NO
YES
I=I+1
TIME=TIME+DTSIM
RETURN

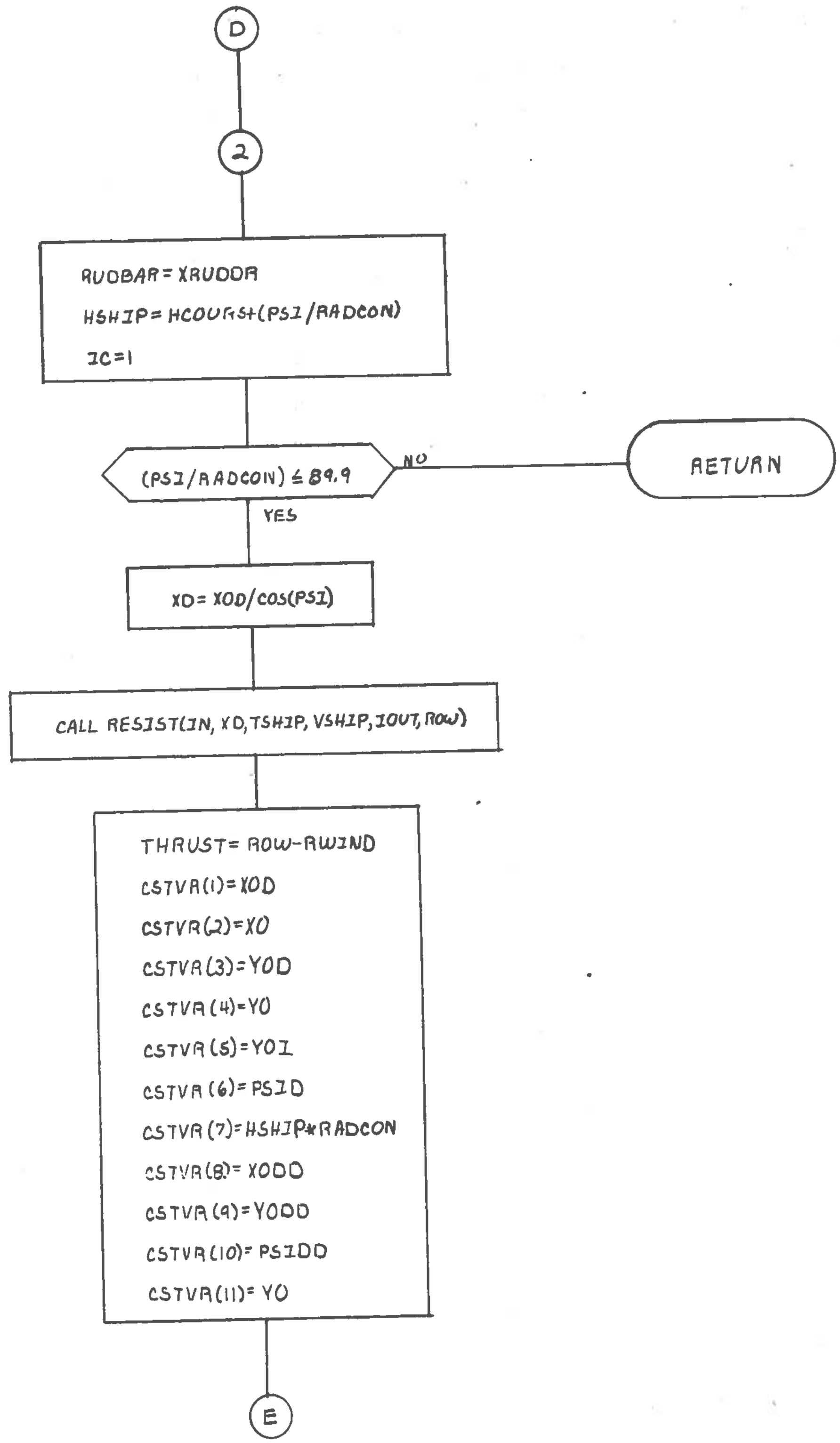
```
SUBROUTINE  SSCON(IOUT, CSTVR, SAMP)

COMMON/CONST/RHOA, RHOW, GRAV, PI, RADCON, DISPLC, FPSMPH
COMMON/SIM/ TSTART, XSTART, XEND, TPRT, TDATA, DTSIM, DTPRT,
      DTDATA, IANG, NANG
COMMON/REACH/COURSE(10,3), NCOURS, EDGE(10,3), NEDGE, XCURNT(10,3),
      NCURNT, WDEPTH, YCSS, HCSS
COMMON/WEATHR/VCURNT, HCURNT, XWIND(10,3), NWIND
COMMON/CONTRL/CPOS(4), CHEAD(3), XRUDSS, PSISS, RUDBAR
COMMON/COEFM/XMCOEF(15), XUD, YVD, YV, YPSIDD, YPSID, XNPSDD,
      XNVD, XNPSID, XNV, YRUD, XNRUD
COMMON/SHIP/VSHIP, TSHIP, XLBP, XLWL, XLOA, BEAM, DRAFT,
      AREAP, AREAF DISPL, XIZ, RUDMAX
COMMON/COEFCH/XYOP(7,11,2), XWPB(7), NYPB(7), XALPHA(11,3),
      NWPB, NXALP

DIMENSION  CSTVR(15), SAMP(19)
```

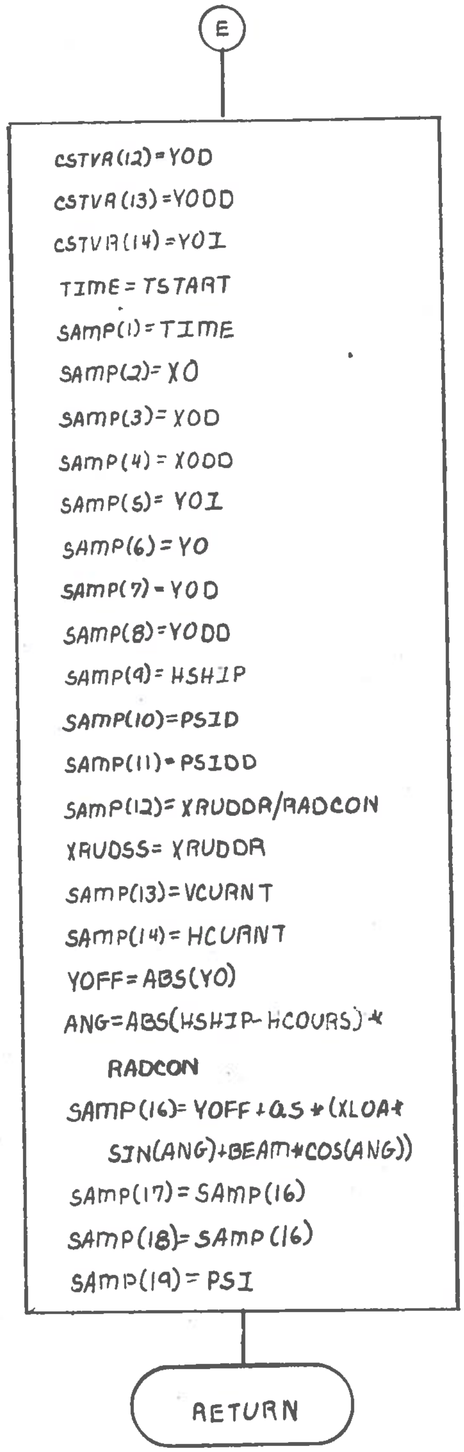


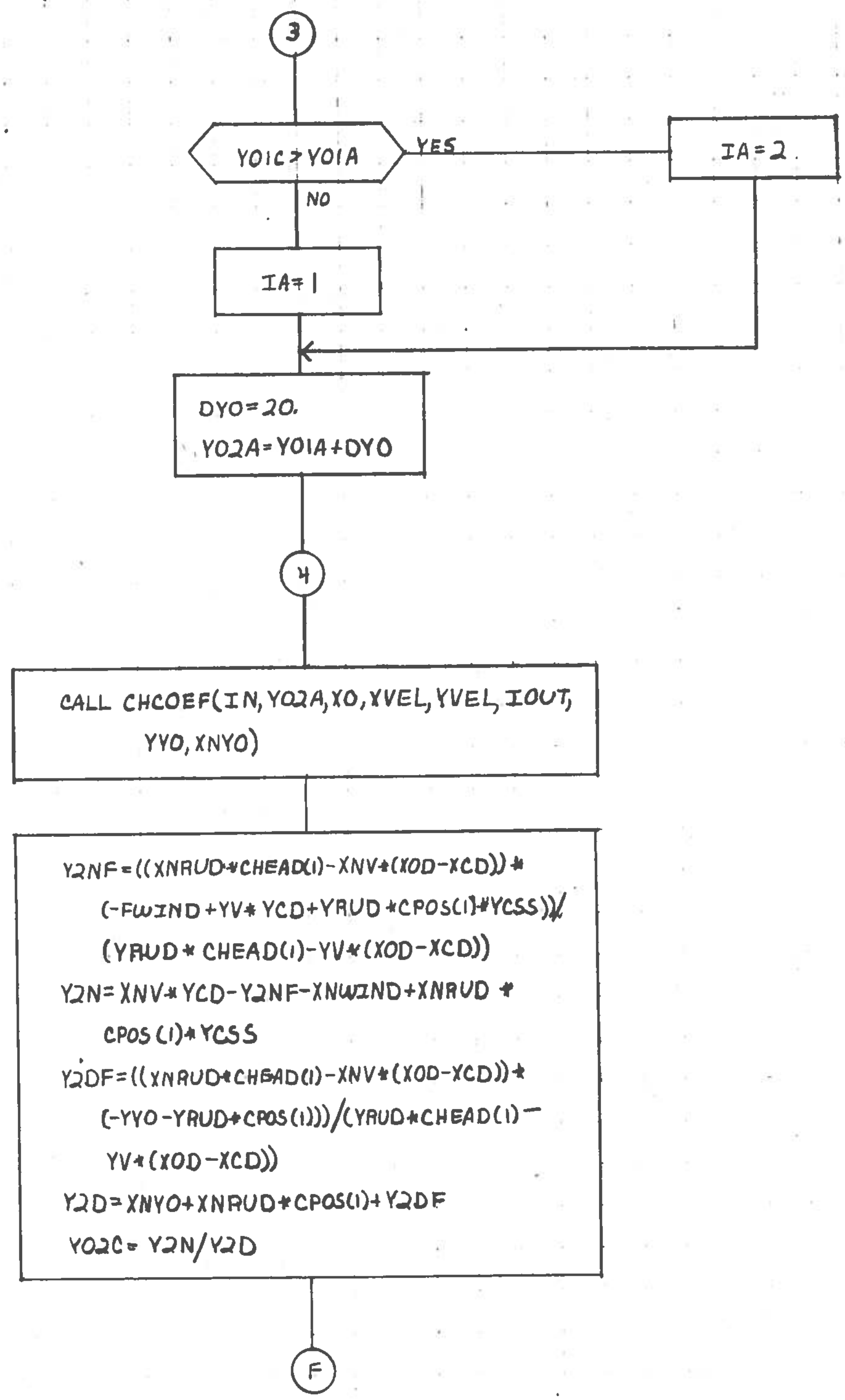

A
YO=YCOURS
YCSS=YCOURS
HCSS=HCOURS
YOD=0.
YODD=0.
YOI=0.
PSID=0.
PSIDD=0.
YOIA=YCOURS
IC=0
ID=0
CALL INTER(IN, IANG, XCURNT, NCURNT, XO, IOUT, VCURNT, HCURNT)
HCR=HCURNT*RADCON
HSR=HCSS*RADCON
XCD=VCURNT*(SIN(HCR)*SIN(HSR)+COS(HCR)*COS(HSR))
YCD=VCURNT*(SIN(HCR)*COS(HSR)-COS(HCR)*SIN(HSR))
CALL WIND(IN, TSTART, HSHIP, XOD, YOD, IOUT, RWIND, FWIND, XNWIND)
B

```
(B)

XVEL= XOD-XCD
YVEL= YOD-YCD
VEL= SQRT(XVEL*XVEL+YVEL*YVEL)
YVD= 0.5*XMCOEF(1)*RHOW*XLWL*XLWL*XLWL
YV= 0.5*XMCOEF(2)*RHOW*XLWL*XLWL*VEL
YPSIDD= 0.5*XMCOEF(4)*RHOW*XLWL**4.
YPSID=0.5*XMCOEF(5)*RHOW*VEL*XLWL**3.
XNPSDD=0.5*XMCOEF(6)*RHOW*XLWL**5.
XNVD=0.5*XMCOEF(7)*RHOW*XLWL**4.
XNPSID=0.5*XMCOEF(8)*RHOW*VEL*XLWL**4.
XNV=0.5*XMCOEF(9)*RHOW*VEL*XLWL**3
[illegible]RUD=0.5*[illegible]COEF(11[illegible]OW*VEL*VEL*
    XLWL*XLWL
XNRUD=0.5*XMCOEF(12)*RHOW*VEL*VEL*
    XLWL**3.
XUD=XMCOEF(13)*DISPLC*DISPL/GRAV

CALL CHCOEF(IN,YOIA,XO,XVEL,YVEL,IOUT,
    YYO, XNYO)

|CPOS(I)| > 10**-20 ?
  NO  -> (C)
  YES ->
    PN=(XOD-XCD)*(YRUD/XNRUD*XNV-YV)
    D1=(YRUD/XNRUD)*(XNWIND+XNYO*YO+
        YCD*XNV)
    D2=-FWIND-YYO*YO-YOC*YV
    PSI=(D1+D2)/PN
    XRUDDR=(XNV*(XOC-XCD)*PSI-XNWIND-
        XNYO*YO-YCD*XNV)/XNRUD
    PSISS= PSI
    XRUDSS=XRUDDR
    -> (2)
```

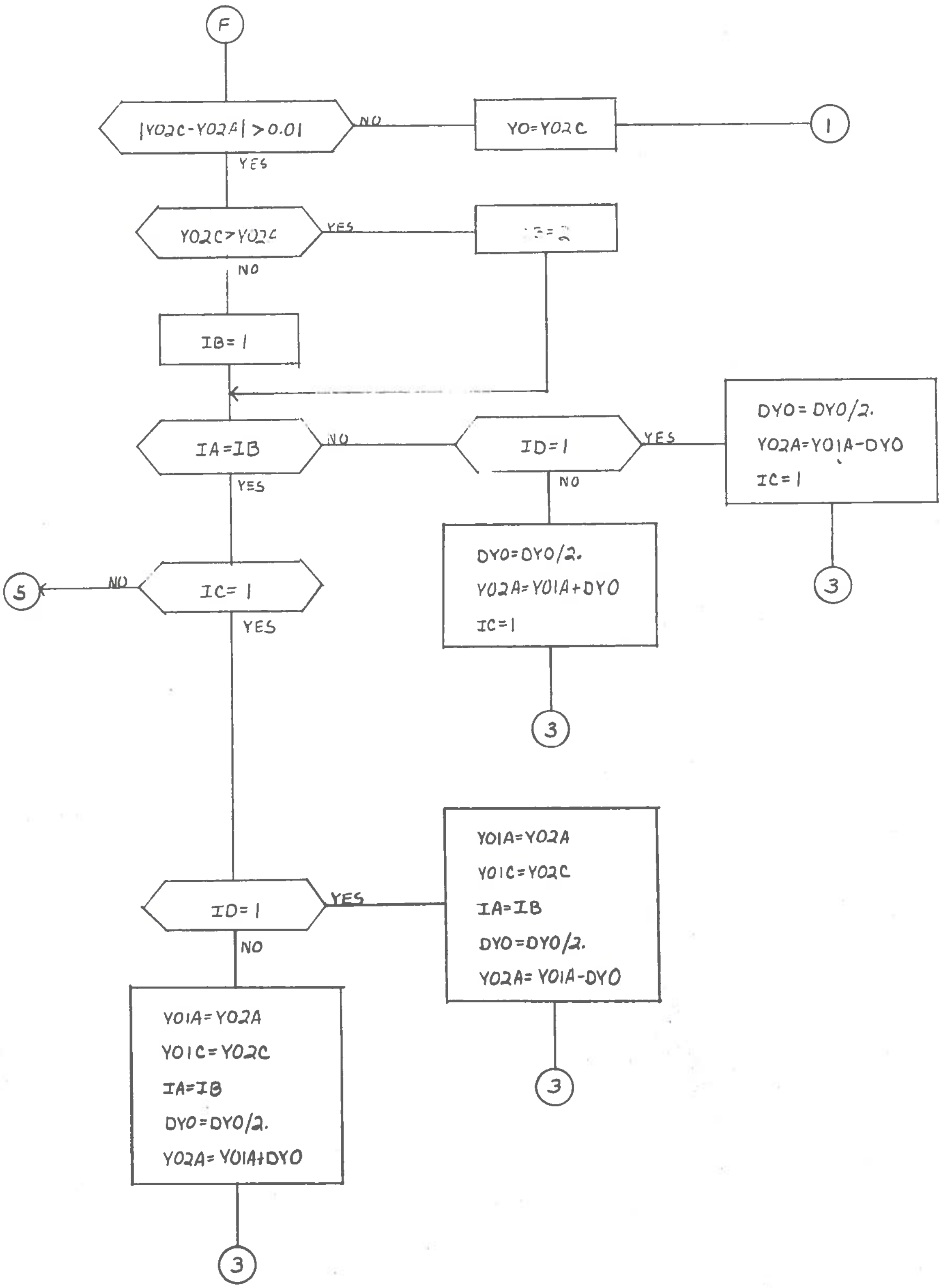

C
YINF=((XNRUD*CHEAD(I)-XNV*(XOD-XCD))*(-FWIND+
YV*YCD+YRUD*CPOS(I)*YCSS))/(YRUD*CHEAD(I)-
YV*(XOD-XCD))
YIN=XNV*YCD-YINF-XNWIND+XNRUD*CPOS(I)*YCSS
YIDF=((XNRUD*CHEAD(I)-XNV*(XOD-XCD))*(-YYO-
YRUD*CPOS(I)))/(YRUD*CHEAD(I)-YV*
(XOD-XCD))
YID=XNYO+XNRUD*CPOS(I)+YIDF
YOIC=YIN/YID
|YOIC-YOIA|>0.01
YES
3
NO
YO=YOIA
1
PSISS=0.
XRUDSS=0.
PSI=(-FWIND+YV*YCD-YO*(YYO+YRUD*CPOS(I))+
YRUD*CPOS(I)*YCSS)/(YRUD*CHEAD(I)-YV*
(XOD-XCD))
XRUDDR=CPOS(I)*(YO-YCSS)+CHEAD(I)*(PSI-PSISS)
+XRUDSS
D

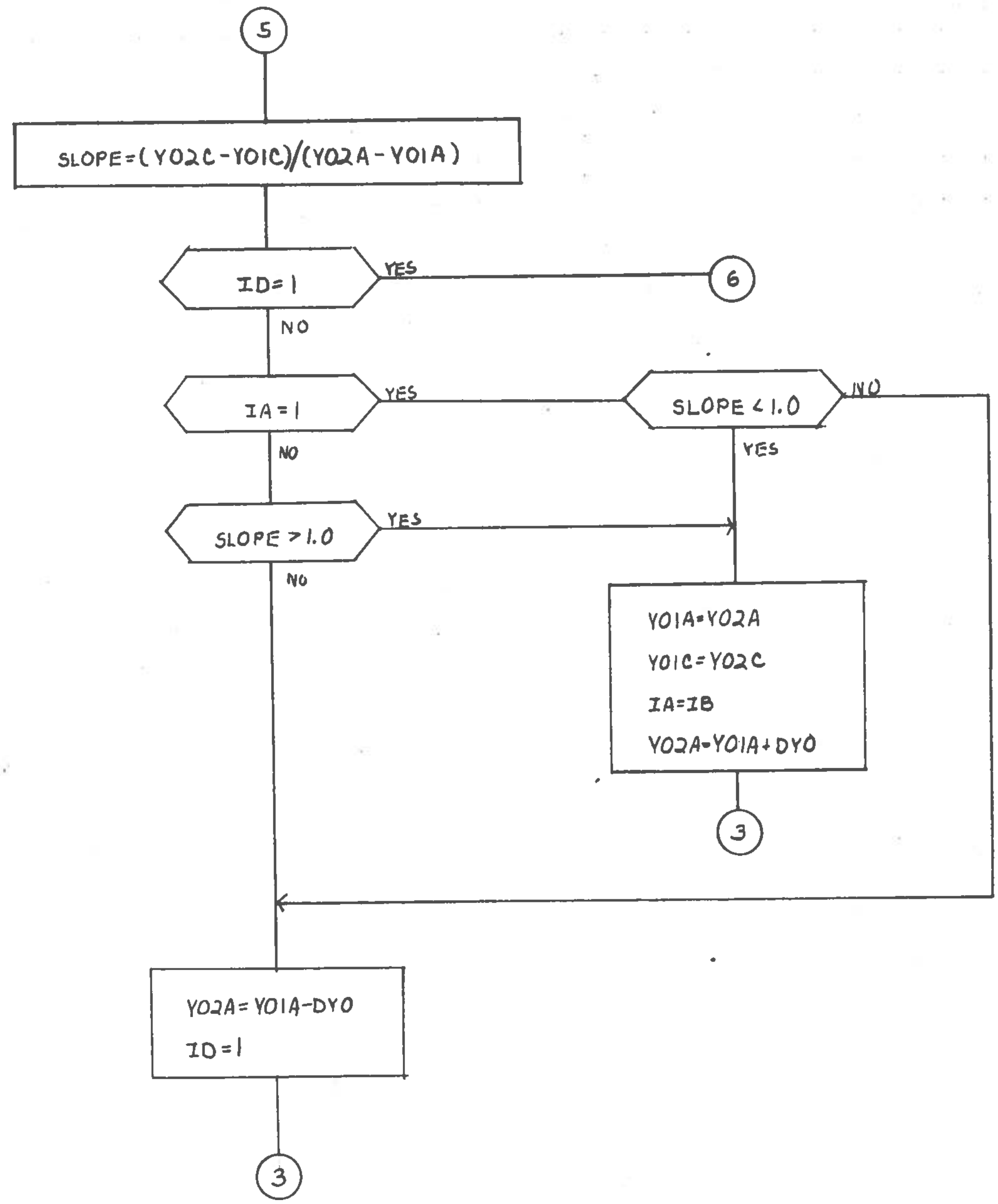

D
2
RUDBAR=XRUDDR
HSHIP=HCOURS+(PSI/RADCON)
IC=1
(PSI/RADCON) ≤ 89.9
NO
RETURN
YES
XD=XOD/COS(PSI)
CALL RESIST(IN,XD,TSHIP,VSHIP,IOUT,ROW)
THRUST=ROW-RWIND
CSTVR(1)=XOD
CSTVR(2)=XO
CSTVR(3)=YOD
CSTVR(4)=YO
CSTVR(5)=YOI
CSTVR(6)=PSID
CSTVR(7)=HSHIP*RADCON
CSTVR(8)=XODD
CSTVR(9)=YODD
CSTVR(10)=PSIDD
CSTVR(11)=YO
E

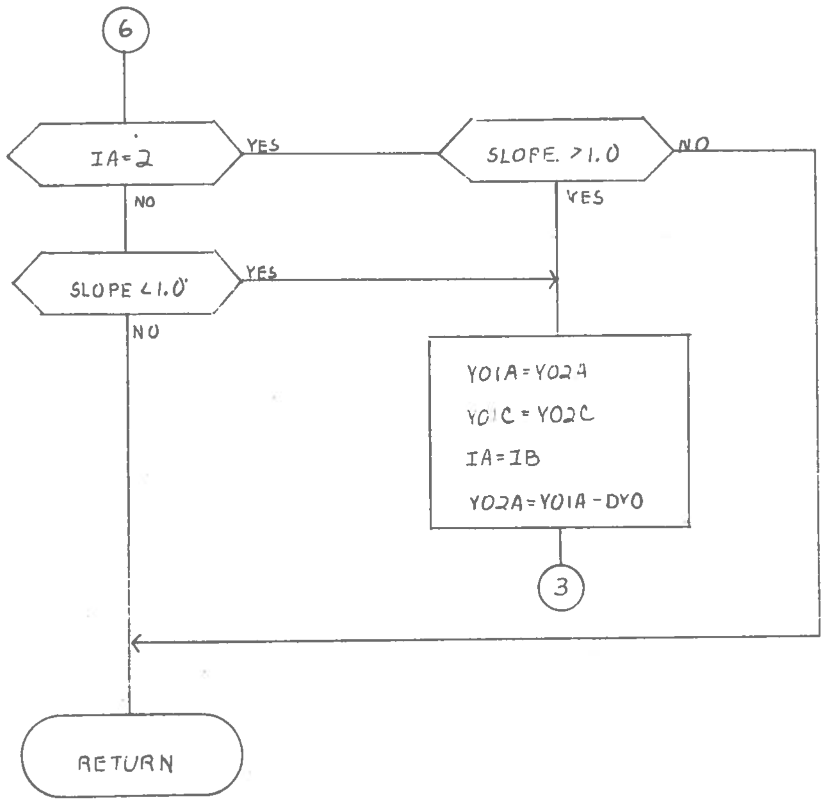

E
CSTVR(12)=YOD
CSTVR(13)=YODD
CSTVR(14)=YOI
TIME=TSTART
SAMP(1)=TIME
SAMP(2)=XO
SAMP(3)=XOD
SAMP(4)=XODD
SAMP(5)=YOI
SAMP(6)=YO
SAMP(7)=YOD
SAMP(8)=YODD
SAMP(9)=HSHIP
SAMP(10)=PSID
SAMP(11)=PSIDD
SAMP(12)=XRUDDR/RADCON
XRUDSS=XRUDDR
SAMP(13)=VCURNT
SAMP(14)=HCURNT
YOFF=ABS(YO)
ANG=ABS(HSHIP-HCOURS)*
RADCON
SAMP(16)=YOFF+0.5*(XLOA*
SIN(ANG)+BEAM*COS(ANG))
SAMP(17)=SAMP(16)
SAMP(18)=SAMP(16)
SAMP(19)=PSI
RETURN



3
Y01C > Y01A
YES
IA=2
NO
IA=1
DYO=20.
Y02A=Y01A+DYO
4
CALL CHCOEF(IN,Y02A,XO,XVEL,YVEL,IOUT,
YYO,XNYO)
Y2NF=((XNRUD*CHEAD(I)-XNV*(XOD-XCD))*
(-FWIND+YV*YCD+YRUD*CPOS(I)*YCSS))/
(YRUD*CHEAD(I)-YV*(XOD-XCD))
Y2N=XNV*YCD-Y2NF-XNWIND+XNRUD*
CPOS(I)*YCSS
Y2DF=((XNRUD*CHEAD(I)-XNV*(XOD-XCD))*
(-YYO-YRUD*CPOS(I)))/(YRUD*CHEAD(I)-
YV*(XOD-XCD))
Y2D=XNYO+XNRUD*CPOS(I)+Y2DF
Y02C=Y2N/Y2D
F

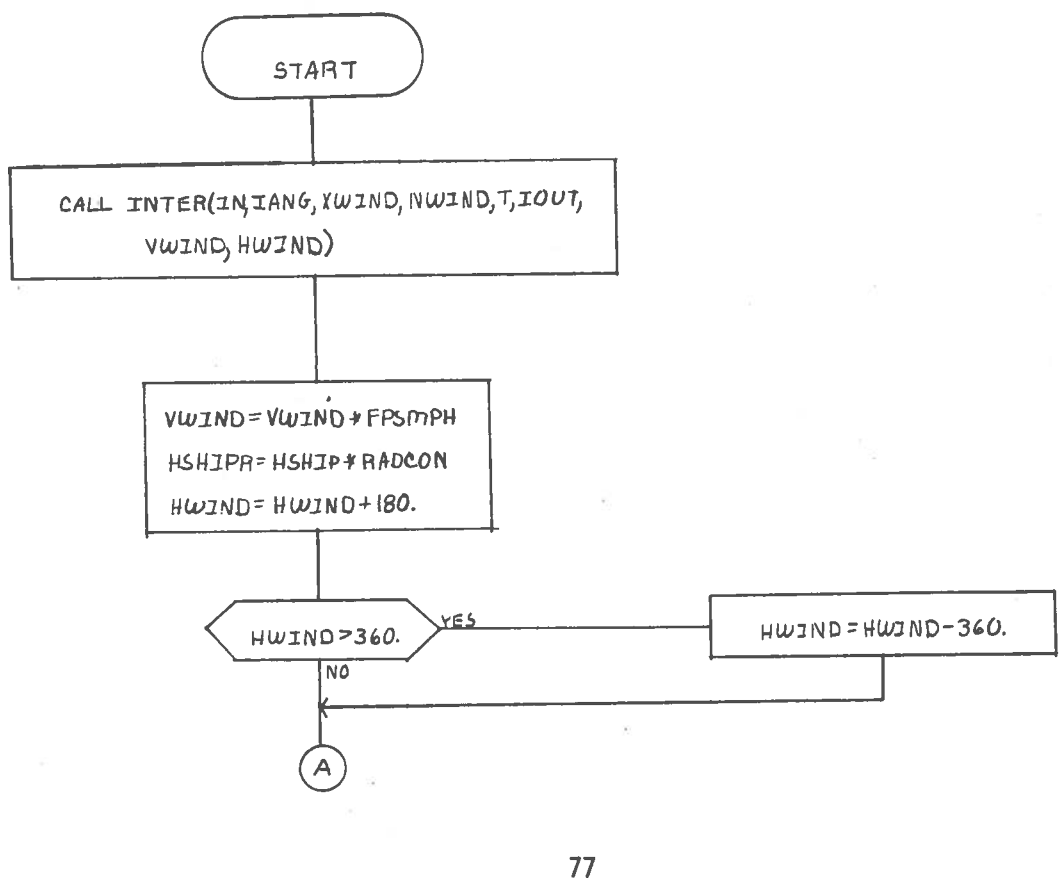

F
|YO2C-YO2A| > 0.01
NO
YO=YO2C
1
YES
YO2C>YO2C
YES
NO
IB=1
IA=IB
NO
ID=1
YES
DYO=DYO/2.
YO2A=YO1A-DYO
IC=1
NO
DYO=DYO/2.
YO2A=YO1A+DYO
IC=1
3
3
YES
NO
IC=1
5
YES
ID=1
YES
YO1A=YO2A
YO1C=YO2C
IA=IB
DYO=DYO/2.
YO2A=YO1A-DYO
3
NO
YO1A=YO2A
YO1C=YO2C
IA=IB
DYO=DYO/2.
YO2A=YO1A+DYO
3

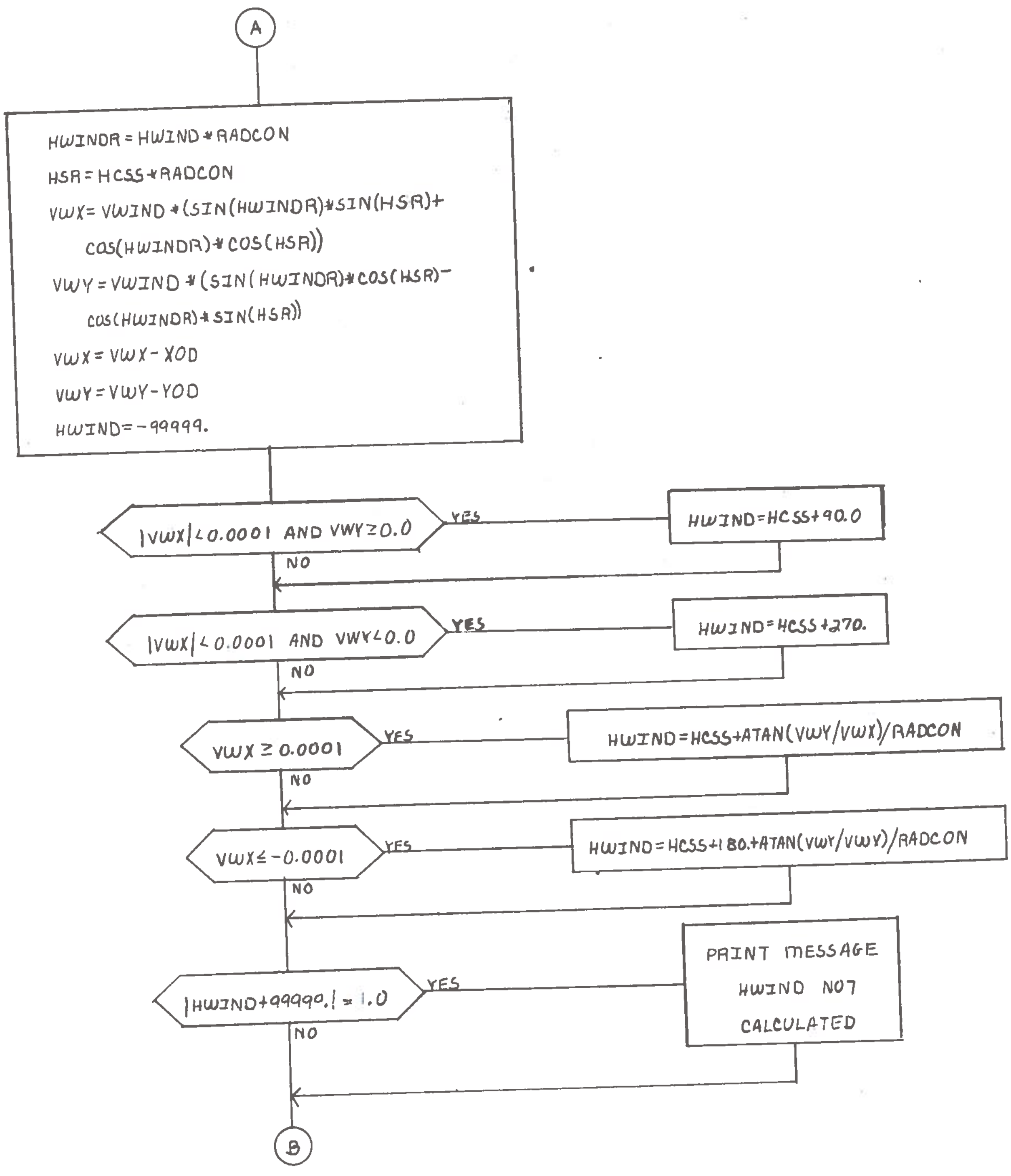

5
SLOPE=(YO2C-YO1C)/(YO2A-YO1A)
ID=1
YES
6
NO
IA=1
YES
SLOPE < 1.0
NO
YES
NO
SLOPE > 1.0
YES
NO
YO1A=YO2A
YO1C=YO2C
IA=IB
YO2A=YO1A+DYO
3
YO2A=YO1A-DYO
ID=1
3

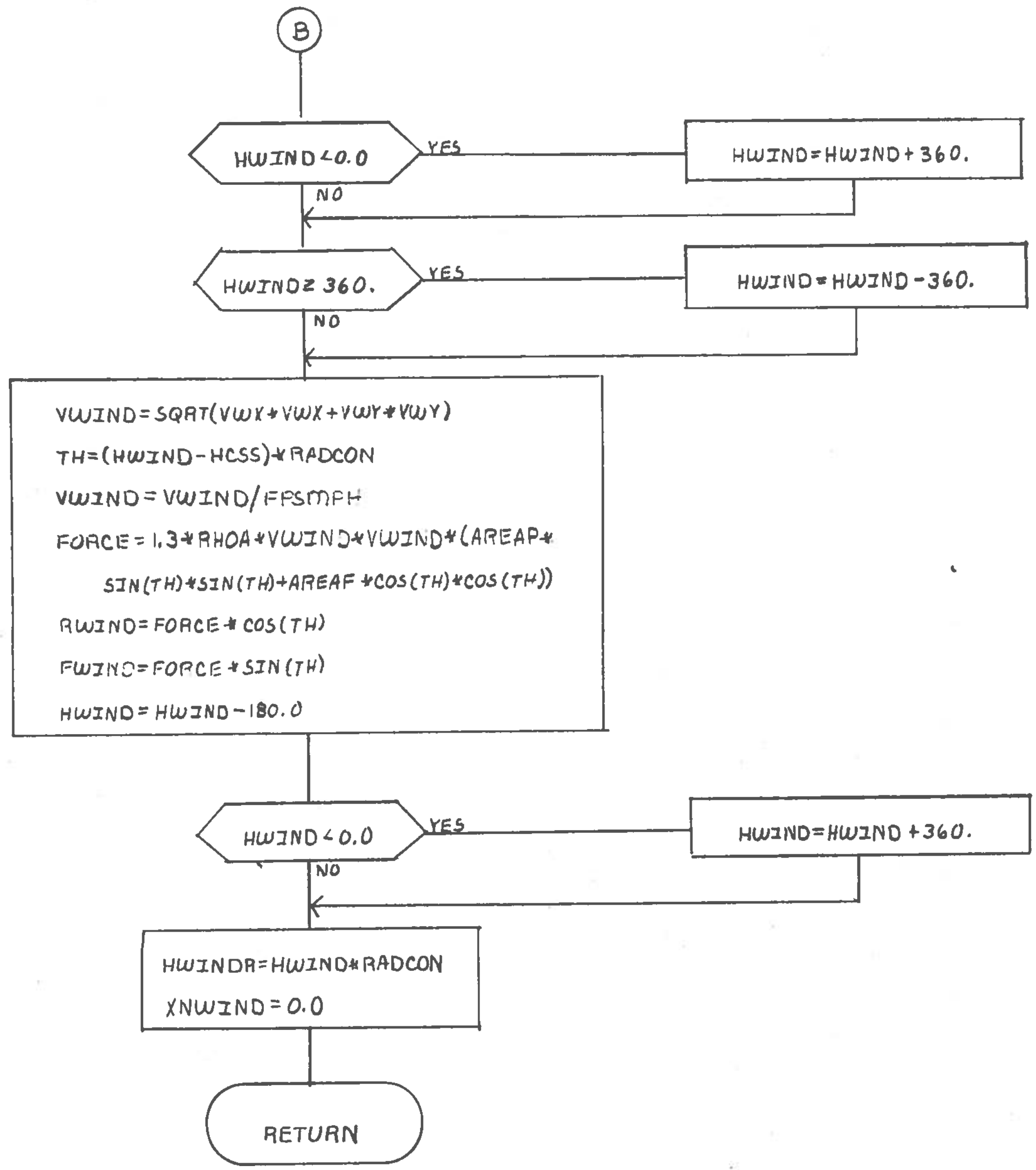

6
IA=2
YES
NO
SLOPE > 1.0
NO
YES
SLOPE < 1.0
YES
NO
Y01A=Y02A
Y01C=Y02C
IA=IB
Y02A=Y01A-DY0
3
RETURN

SUBROUTINE WIND(IN, T, HSHIP, YOD, YOD, IOUT, RWIND, FWIND, XNWIND)

```
COMMON/CONST/RHOA, RHOW, GRAV, PI, RADCON, DISPLC, FPSMPH
COMMON/SIM/TSTART, XSTART, XEND, TPRT, TDATA, DTSIM, DTPRT,
   DTDATA, IANG, NANG
COMMON/REACH/COURSE(10,3), NCOURS, EDGE(10,3), NEDGE, XCURNT(10,3),
   NCURNT, WDEPTH, YCSS, HCSS
COMMON/WEATHR/VCURNT, HCURNT, XWIND(10,3), NWIND
COMMON/CONTRL/CPOS(4), CHEAD(3), XRUDSS, PSISS, RUDBAR
COMMON/COEFM/XMCOEF(15), XUD, YVD, YV, YPSIDD, YPSID, XNPSDD,
   XNVD, XNPSID, XNV, YRUD, YNRUD
COMMON/SHIP/VSHIP, TSHIP, XLBP, YLWL, XLOA, BEAM, DRAFT,
   AREAP, AREAF, DISPL, XIZ, RUDMAX
COMMON/COEFCH/XYOP(7,11,2), XWPB(7), NYPB(7), XALPHA(11,3),
   NWPB, NXALP
```




A
HWINDR = HWIND * RADCON
HSR = HCSS * RADCON
VWX = VWIND * (SIN(HWINDR) * SIN(HSR) + COS(HWINDR) * COS(HSR))
VWY = VWIND * (SIN(HWINDR) * COS(HSR) - COS(HWINDR) * SIN(HSR))
VWX = VWX - XOD
VWY = VWY - YOD
HWIND = -99999.
|VWX| < 0.0001 AND VWY ≥ 0.0
YES
HWIND = HCSS + 90.0
NO
|VWX| < 0.0001 AND VWY < 0.0
YES
HWIND = HCSS + 270.
NO
VWX ≥ 0.0001
YES
HWIND = HCSS + ATAN(VWY/VWX)/RADCON
NO
VWX ≤ -0.0001
YES
HWIND = HCSS + 180. + ATAN(VWY/VWX)/RADCON
NO
|HWIND + 99999.| ≤ 1.0
YES
PRINT MESSAGE HWIND NOT CALCULATED
NO
B


B
HWIND<0.0
YES
HWIND=HWIND+360.
NO
HWIND≥360.
YES
HWIND=HWIND-360.
NO
VWIND=SQRT(VWX*VWX+VWY*VWY)
TH=(HWIND-HCSS)*RADCON
VWIND=VWIND/FPSMPH
FORCE=1.3*RHOA*VWIND*VWIND*(AREAP*
SIN(TH)*SIN(TH)+AREAF*COS(TH)*COS(TH))
RWIND=FORCE*COS(TH)
FWIND=FORCE*SIN(TH)
HWIND=HWIND-180.0
HWIND<0.0
YES
HWIND=HWIND+360.
NO
HWINDR=HWIND*RADCON
XNWIND=0.0
RETURN

C.6 LISTING OF PROGRAM MANVER

```
*        PROGRAM MANVER 10/6/77
*
*        COMPUTER PROGRAM TO SIMULATE THE MANEUVERING OF A SHIP
*        ALONG A COURSE IN A SHALLOW, RESTRICTED CHANNEL IN THE
*        PRESENCE OF EXTERNAL FORCES AND MOMENTS DUE TO WIND
*        AND CURRENT
*
        COMMON/CONST/RHOA,RHOW,GRAV,PI,RADCON,DISPLC,FPSMPH
        COMMON/SIM/TSTART,XSTART,XEND,TPRT,TDATA,DTSIM,DTPRT,
+        DTDATA,IANG,NANG
        COMMON/REACH/COURSE(10,3),NCOURS,EDGE(10,3),NEDGE,
+        XCURNT(10,3),NCURNT,WDEPTH,YCSS,HCSS
        COMMON/WEATHR/VCURNT,HCURNT,XWIND(10,3),NWIND
        COMMON/CONTRL/CPOS(4),CHEAD(3),XRUDSS,PSISS,RUDBAR
        COMMON/COEFM/XMCOEF(15),XUD,YVD,YV,YPSIDD,YPSID,XNPSDD,
+       XNVD,XNPSID,XNV,YRUD,XNRUD
        COMMON/SHIP/VSHIP,TSHIP,XLBP,XLWL,XLOA,BEAM,DRAFT,AREAP,
+       AREAF,DISPL,XIZ,RUDMAX
        COMMON/COEFCH/XYOP(7,11,2),XWPB(7),NYPB(7),XALPHA(11,3),
+        NWPB,NXALP
        DIMENSION CSTVR(15),SAMP(19)
*
        CALL DEFINE(5,'SIMDTA,')
        CALL DEFINE(6,'ADSIM,')
        CALL DEFINE(1,'OUT384,')
*
    999 CONTINUE
        REWIND 6
        IF(IEOF(5).EQ.1)GO TO 1000
*        DEFINE CONSTANTS
*
        PI=3.14159
        RADCON=PI/180.
        FPSMPH=1.467
        IANG=1
        NANG=0
*
*        INITIALIZE ARRAYS
*
        DO 1 I=1,15
      1 CSTVR(I)=0.0
        DO 2 I=1,19
      2 SAMP(I)=0.0
*
*
*        ENTER INPUT DATA
*
        CALL INDAT
*
*        ESTABLISH STEADY STATE CONDITIONS
*
        CALL SSCON(IOUT,CSTVR,SAMP)
*
*        PRINT HEADINGS AND STEADY STATE CONDITIONS
*
        CALL PRT(IN,SAMP)
*
```

```
*         INTIALIZE SIMULATION, PRINT AND DATA UPDATE TIMES
*
          TIME=TSTART
          TPRT=TSTART+DTPRT
          TDATA=TSTART+DTDATA
*
*         STEP THROUGH TIME FROM TSTART TO TEND IN INCREMENTS
*         OF DTSIM INTEGRATING DIFFERENTIAL EQUATION OF MOTION
*         DESCRIBING MANEUVERING
*
      100 CONTINUE
          NVAR=7
*
          CALL SOLVE(IN,NVAR,TIME,IOUT,CSTVR)
*
*         ESTABLISH POSITION OF SHIP RELATIVE TO NAVIGATION
*
*         CHANNEL
          XOD=CSTVR(1)
          XO=CSTVR(2)
          YOD=CSTVR(3)
          YO=CSTVR(4)
          YOI=CSTVR(5)
          PSID=CSTVR(6)
          HSHIPR=CSTVR(7)
          HSHIP=HSHIPR/RADCON
          CALL INTER(IN,IANG,COURSE,MCOURS,XO,IOUT,YCOURS,HCOURS)
*
*
          SAMP(1)=TIME
          SAMP(2)=XO
          SAMP(3)=XOD
          SAMP(5)=YOI
          SAMP(6)=YO
          SAMP(7)=YOD
          HSHIP=HSHIPR/RADCON
          SAMP(9)=HSHIP
          SAMP(10)=PSID/RADCON
          XRUDDR=CSTVR(15)
          SAMP(12)=XRUDDR/RADCON
          SAMP(13)=VCURNT
          SAMP(14)=HCURNT
          YOFF=ABS(YO)
          ANG=ABS(HSHIP-HCOURS)*RADCON
          SAMP(16)=YOFF+0.5*(XLOA*SIN(ANG)+BEAM*COS(ANG))
          IF(SAMP(16).GT.SAMP(17))SAMP(17)=SAMP(16);SAMP(15)=SAMP(2)
          SAMP(19)=PSI
*
*         PRINT SIMULATION OUTPUT IF TIME IS EQUAL TO OR GREATER
*         THAN TPRT
*
          IF(TIME.LT.(TPRT-0.0001))GO TO 150
*
          CALL PRT(IN,SAMP)
*
          TPRT=TPRT+DTPRT
      150 CONTINUE
```

```
*
*     UPDATA RUDDER DATA ON POSITION IF TIME IS GREATER
*     THAN TDATA
*
*     IF(TIME.LT.(TDATA-0.0001))GO TO 160
      CSTVR(11)=YO
      CSTVR(12)=YOD
      CSTVR(13)=YODD
      CSTVR(14)=YOI
*     TDATA=TDATA+DTDATA
  160 CONTINUE
*
*      END SIMULATION IF TIME IS GREATER THAN TEND
*
      IF(CSTVR(2).LE.XEND)GO TO 100
  180 CONTINUE
*
*      PRINT SUMMARY OF SIMULATION RESULTS
*
*
      WRITE(1,170)SAMP(17),SAMP(15)
  170 FORMAT(/'MAX. EXCURSION FROM CHANNEL CENTERLINE =',F10.2,
     +' AT XO =',F10.2/////)
      GO TO 999
 1000 CONTINUE
      STOP
      END
      SUBROUTINE INDAT
*
*      PURPOSE:  TO READ INPUT DATA FROM DATA FILE
*
      COMMON/CONST/RHOA,RHOW,GRAV,PI,RADCON,DISPLC,FPSMPH
      COMMON/SIM/TSTART,XSTART,XEND,TPRT,TDATA,DTSIM,DTPRT,
     + DTDATA,IANG,NANG
      COMMON/REACH/COURSE(10,3),NCOURS,EDGE(10,3),NEDGE,
     + XCURNT(10,3),NCURNT,WDEPTH,YCSS,HCSS
      COMMON/WEATHR/VCURNT,HCURNT,XWIND(10,3),NWIND
      COMMON/CONTRL/CPOS(4),CHEAD(3),XRUDSS,PSISS,RUDBAR
      COMMON/COEFM/XMCOEF(15),XUD,YVD,YV,YPSIDD,YPSID,XNPSDD,
     + XNVD,XNPSID,XNV,YRUD,XNRUD
      COMMON/SHIP/VSHIP,TSHIP,XLBP,XLWL,XLOA,BEAM,DRAFT,AREAP,
     + AREAF,DISPL,XIZ,RUDMAX
      COMMON/COEFCH/XYOP(7,11,2),XWPB(7),NYPB(7),XALPHA(11,3),
     + NWPB,NXALP
      DIMENSION ITITLE(30)
*
*      EACH GROUPING IN THE DATA FILE IS PRECEEDED BY A
*      LINE THAT MUST BE READ AS A BLANK LINE
*
*     READ IN RUN NUMBER
      READ(5,3)NRUN1,NRUN2
      WRITE(9,3)NRUN1,NRUN2
    3 FORMAT(2A2)
*
      CALL DATE(IMON,IDAY,IYER)
*
      WRITE(1,4)IMON,IDAY,IYER,NRUN1,NRUN2
```

```
   4 FORMAT(////29X,'MANEUVERING SIMULATION'/36X,A2,2('/',A2)/
    +37X,'RUN ',2A2///35X,'INPUT DATA'/)
*      READ CONSTANTS
     READ(5,5)(ITITLE(I),I=1,30)
   5 FORMAT(30A2)
     WRITE(1,5)(ITITLE(I),I=1,30)
     READ(5,10)RHOA,RHOW,GRAV,DISPLC
     WRITE(1,10)RHOA,RHOW,GRAV,DISPLC
  10 FORMAT(4F10.2)
*
*      READ IN SIMULATION DATA
     READ(5,5)(ITITLE(I),I=1,30)
     WRITE(1,5)(ITITLE(I),I=1,30)
     READ(5,20)TSTART,XSTART,XEND
     WRITE(1,20)TSTART,XSTART,XEND
  20 FORMAT(3F10.2)
     READ(5,5)(ITITLE(I),I=1,30)
     WRITE(1,5)(ITITLE(I),I=1,30)
     READ(5,20)DTSIM,DTPRT,DTDATA
     WRITE(1,20)DTSIM,DTPRT,DTDATA
*
*      READ REACH DATA
     READ(5,5)(ITITLE(I),I=1,30)
     WRITE(1,5)(ITITLE(I),I=1,30)
     READ(5,25)NCOURS,NEDGE,NCURNT,WDEPTH
     WRITE(1,25)NCOURS,NEDGE,NCURNT,WDEPTH
  25 FORMAT(3I5,F10.2)
*
*      READ COURSE ARRAY
     READ(5,5)(ITITLE(I),I=1,30)
     WRITE(1,5)(ITITLE(I),I=1,30)
     DO 30 I=1,NCOURS
     READ(5,20)(COURSE(I,J),J=1,3)
     WRITE(1,20)(COURSE(I,J),J=1,3)
  30 CONTINUE
*
*      READ EDGE ARRAY
     READ(5,5)(ITITLE(I),I=1,30)
     WRITE(1,5)(ITITLE(I),I=1,30)
     DO 35 I=1,NEDGE
     READ(5,20)(EDGE(I,J),J=1,3)
     WRITE(1,20)(EDGE(I,J),J=1,3)
  35 CONTINUE
*
*      READ XCURNT ARRAY
     READ(5,5)(ITITLE(I),I=1,30)
     WRITE(1,5)(ITITLE(I),I=1,30)
     DO 40 I=1,NCURNT
     READ(5,20)(XCURNT(I,J),J=1,3)
     WRITE(1,20)(XCURNT(I,J),J=1,3)
  40 CONTINUE
*
*      READ WIND DATA
     READ(5,5)(ITITLE(I),I=1,30)
     WRITE(1,5)(ITITLE(I),I=1,30)
     READ(5,45)NWIND
     WRITE(1,45)NWIND
```

```
   45 FORMAT(I2)
      READ(5,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      DO 50 I=1,NWIND
      READ(5,20)(XWIND(I,J),J=1,3)
      WRITE(1,20)(XWIND(I,J),J=1,3)
   50 CONTINUE
*
*       READ CONTROL DATA
      READ(5,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(5,51)(CPOS(I),I=1,4)
      WRITE(1,51)(CPOS(I),I=1,4)
   51 FORMAT(3F12.7,F15.13)
      READ(5,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(5,51)(CHEAD(I),I=1,3)
      WRITE(1,51)(CHEAD(I),I=1,3)
*
*       READ SHIP MANEUVERING COEFFICIENTS
*
      READ(5,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(5,55)(XMCOEF(I),I=1,15)
      WRITE(1,55)(XMCOEF(I),I=1,15)
   55 FORMAT(5F12.5)
*
*       READ SHIP DATA
      READ(5,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(5,61)DISPL,BEAM,DRAFT,XIZ
      WRITE(1,61)DISPL,BEAM,DRAFT,XIZ
   61 FORMAT(3F10.2,F20.0)
   60 FORMAT(6F10.2)
      READ(5,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(5,60)XLBP,XLWL,XLOA
      WRITE(1,60)XLBP,XLWL,XLOA
      READ(5,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(5,60)VSHIP,TSHIP,AREAP,AREAF
      VSHIP=VSHIP*FPSMPH
      WRITE(1,60)VSHIP,TSHIP,AREAP,AREAF
      READ(5,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(5,60)RUDMAX
      RUDMAX=RUDMAX*RADCON
      WRITE(1,60)RUDMAX
*
*       READ NWPB AND NXALP
      READ(6,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(6,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(6,75)NWPB,NXALP
   75 FORMAT(2I5)
      WRITE(1,75)NWPB,NXALP
```

```
*      READ XWPB ARRAY
      READ(6,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(6,80)(XWPB(I),I=1,NWPB)
   80 FORMAT(7F10.2)
      WRITE(1,80)(XWPB(I),I=1,NWPB)
*      READ NYPB ARRAY
      READ(6,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(6,85)(NYPB(I),I=1,NWPB)
   85 FORMAT(7I5)
      WRITE(1,85)(NYPB(I),I=1,NWPB)
*      READ XYOP ARRAY
      READ(6,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      DO 95 I=1,NWPB
      JT=NYPB(I)
      READ(6,90)(XYOP(I,J,1),J=1,JT)
      WRITE(1,90)(XYOP(I,J,1),J=1,JT)
   90 FORMAT(6F10.3)
      READ(6,90)(XYOP(I,J,2),J=1,JT)
      WRITE(1,90)(XYOP(I,J,2),J=1,JT)
   95 CONTINUE
*      READ XALPHA ARRAY
      READ(6,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      READ(6,5)(ITITLE(I),I=1,30)
      WRITE(1,5)(ITITLE(I),I=1,30)
      DO 99 I=1,NXALP
      READ(6,97)(XALPHA(I,J),J=1,3)
      WRITE(1,97)(XALPHA(I,J),J=1,3)
   97 FORMAT(3F10.2)
   99 CONTINUE
  100 FORMAT(///31X,'SIMULATION RESULTS'//30X,'SHIP',41X,'DIST'/
     +3X,'TIME',6X,'XO',8X,'YO',2X,'HEADING',2X,'DXO/DT',3X,
     +'DYO/DT',3X,'DPSI/DT',3X,'RUDDER',3X,'FR CL'/
     +2X,'(SEC)',5X,'(FT)',6X,'(FT)',3X,'(DEG)',4X,'(FPS)',
     +4X,'(FPS)',2X,'(RAD/SEC)',2X,'(DEG)',5X,'(FT)')
      WRITE(1,100)
      RETURN
      END
*
      SUBROUTINE SSCON(IOUT,CSTVR,SAMP)
*
*      PURPOSE:  ESTABLISH STEADY STATE CONDITIONS FOR SHIP AT
*      TIME=TSTART AND XO=XSTART
*
      COMMON/CONST/RHOA,RHOW,GRAV,PI,RADCON,DISPLC,FPSMPH
      COMMON/SIM/TSTART,XSTART,XEND,TPRT,TDATA,DTSIM,DTPRT,
+      DTDATA,IANG,NANG
      COMMON/REACH/COURSE(10,3),NCOURS,EDGE(10,3),NEDGE,
+      XCURNT(10,3),NCURNT,WDEPTH,YCSS,HCSS
      COMMON/WEATHR/VCURNT,HCURNT,XWIND(10,3),NWIND
      COMMON/CONTRL/CPOS(4),CHEAD(3),XRUDSS,PSISS,RUDBAR
      COMMON/COEFM/XMCOEF(15),XUD,YVD,YV,YPSIDD,YPSID,XNPSDD,
+     XNVD,XNPSID,XNV,YRUD,XNRUD
      COMMON/SHIP/VSHIP,TSHIP,XLBP,XLWL,XLOA,BEAM,DRAFT,AREAP,
```

```
+      AREAF,DISPL,XIZ,RUDMAX
       COMMON/COEFCH/XYOP(7,11,2),XWPB(7),NYPB(7),XALPHA(11,3),
+       NWPB,NXALP
       DIMENSION CSTVR(15),SAMP(19)
*
*       INITIALIZE XO, XOD, XODD
*
       XO=XSTART
       XOD=VSHIP
       XODD=0.
*
*       DETERMINE DESIRED COURSE POSITION AND HEADING (YCOURS,HCOURS)
*
       CALL INTER(IN,IANG,COURSE,NCOURS,XO,IOUT,YCOURS,HCOURS)
*
*       SHIP IS ASSUMED TO HAVE INTEGRAL FEEDBACK ON POSTION,
*       I.E., EVEN IN PRESENCE OF STEADY EXTERNAL FORCES, THE
*       HELMSMAN CAN MAINTAIN COURSE POSITION
*
       YO=YCOURS
       YCSS=YCOURS
       HCSS=HCOURS
       YOD=0.
       YODD=0.
       YOI=0.
       PSID=0.
       PSIDD=0.
*
*       ASSUME INITIAL VALUE SAME AS YCOURS
       YOIA=YCOURS
*
*       PROPER VALUE HAS NOT BEEN BRACKETED
       IC=0
       ID=0
*
       CALL INTER(IN,IANG,XCURNT,NCURNT,XO,IOUT,VCURNT,HCURNT)
*
       HCR=HCURNT*RADCON
       HSR=HCSS*RADCON
       XCD=VCURNT*(SIN(HCR)*SIN(HSR)+COS(HCR)*COS(HSR))
       YCD=VCURNT*(SIN(HCR)*COS(HSR)-COS(HCR)*SIN(HSR))
*
       CALL WIND(IN,TSTART,HSHIP,XOD,YOD,IOUT,RWIND,FWIND,XNWIND)
*
       XVEL=XOD-XCD
       YVEL=YOD-YCD
       VEL=SQRT(XVEL*XVEL+YVEL*YVEL)
       YVD=0.5*XMCOEF(1)*RHOW*XLWL*XLWL*XLWL
       YV=0.5*XMCOEF(2)*RHOW*XLWL*XLWL*VEL
       YPSIDD=0.5*XMCOEF(4)*RHOW*XLWL**4.
       YPSID=0.5*XMCOEF(5)*RHOW*VEL*XLWL**3.
       XNPSDD=0.5*XMCOEF(6)*RHOW*XLWL**5.
       XNVD=0.5*XMCOEF(7)*RHOW*XLWL**4.
       XNPSID=0.5*XMCOEF(8)*RHOW*VEL*XLWL**4.
       XNV=0.5*XMCOEF(9)*RHOW*VEL*XLWL**3.
       YRUD=0.5*XMCOEF(11)*RHOW*VEL*VEL*XLWL*XLWL
       XNRUD=0.5*XMCOEF(12)*RHOW*VEL*VEL*XLWL**3.
```

```
          XUD=XMCOEF(13)*DISPLC*DISPL/GRAV
*
          CALL CHCOEF(IN,YO1A,XO,XVEL,YVEL,IOUT,YYO,XNYO)
*
          IF(ABS(CPOS(4)).GT.10.**-20)GO TO 400
          Y1NF=((XNRUD*CHEAD(1)-XNV*(XOD-XCD))*(-FWIND+YV*YCD+YRUD*
+          CPOS(1)*YCSS))/(YRUD*CHEAD(1)-YV*(XOD-XCD))
          Y1N=XNV*YCD-Y1NF-XNWIND+XNRUD*CPOS(1)*YCSS
          Y1DF=((XNRUD*CHEAD(1)-XNV*(XOD-XCD))*(-YYO-YRUD*CPOS(1)))/
+          (YRUD*CHEAD(1)-YV*(XOD-XCD))
          Y1D=XNYO+XNRUD*CPOS(1)+Y1DF
          YO1C=Y1N/Y1D
          IF(ABS(YO1A-YO1C).GT.0.01)GO TO 190
          YO=YO1C
          GO TO 300
      190 CONTINUE
          IF(YO1C.GT.YO1A)GO TO 194
*         POINT ABOVE DESIRED LINE
          IA=1
          GO TO 197
      194 CONTINUE
*         POINT BELOW DESIRED LINE
          IA=2
      197 CONTINUE
          DYO=20.
          YO2A=YO1A+DYO
      200 CONTINUE
*
*
          CALL CHCOEF(IN,YO2A,XO,XVEL,YVEL,IOUT,YYO,XNYO)
*
          Y2NF=((XNRUD*CHEAD(1)-XNV*(XOD-XCD))*(-FWIND+YV*YCD+YRUD*
+          CPOS(1)*YCSS))/(YRUD*CHEAD(1)-YV*(XOD-XCD))
          Y2N=XNV*YCD-Y2NF-XNWIND+XNRUD*CPOS(1)*YCSS
          Y2DF=((XNRUD*CHEAD(1)-XNV*(XOD-XCD))*(-YYO-YRUD*CPOS(1)))/
+          (YRUD*CHEAD(1)-YV*(XOD-XCD))
          Y2D=XNYO+XNRUD*CPOS(1)+Y2DF
          YO2C=Y2N/Y2D
          IF(ABS(YO2C-YO2A).GT.0.01)GO TO 220
          YO=YO2C
          GO TO 300
      220 CONTINUE
          IF(YO2C.GT.YO2A)GO TO 231
          IB=1
          GO TO 222
      231 CONTINUE
          IB=2
      222 CONTINUE
          IF(IA.EQ.IB)GO TO 230
*         POINT IS BETWEEN VALUES
          IF(ID.EQ.1)GO TO 225
*         FIRST POINT LESS THAN SECOND
          DYO=DYO/2.
          YO2A=YO1A+DYO
          IC=1
          GO TO 200
      225 CONTINUE
```

```
*       FIRST POINT GREATER THAN SECOND
        DYO=DYO/2.
        YO2A=YO1A-DYO
        IC=1
        GO TO 200
    230 CONTINUE
        IF(IC.NE.1)GO TO 240
*       POINT BRACKETED EARLIER
        IF(ID.EQ.1)GO TO 235
        YO1A=YO2A
        YO1C=YO2C
        IA=IB
        DYO=DYO/2.
        YO2A=YO1A+DYO
        GO TO 200
    235 CONTINUE
        YO1A=YO2A
        YO1C=YO2C
        IA=IB
        DYO=DYO/2.
        YO2A=YO1A-DYO
        GO TO 200
    240 CONTINUE
*       BOTH POINTS ON SAME SIDE OF LINE
        SLOPE=(YO2C-YO1C)/(YO2A-YO1A)
        IF(ID.EQ.1)GO TO 250
        IF(IA.EQ.2)GO TO 242
        IF(SLOPE.GT.1.0)GO TO 245
        GO TO 247
    242 CONTINUE
*       ASSUMED VALUES INCREASING
        IF(SLOPE.LT.1.0)GO TO 245
        GO TO 247
    245 CONTINUE
        YO1A=YO2A
        YO1C=YO2C
        IA=IB
        YO2A=YO1A+DYO
        GO TO 200
    247 CONTINUE
        YO2A=YO1A-DYO
        ID=1
        GO TO 200
    250 CONTINUE
        IF(IA.EQ.2)GO TO 252
        IF(SLOPE.LT.1.0)GO TO 255
        GO TO 257
    252 CONTINUE
        IF(SLOPE.GT.1.0)GO TO 255
        GO TO 257
    255 CONTINUE
*       ASSUMED VALUES DECREASING
        YO1A=YO2A
        YO1C=YO2C
        IA=IB
        YO2A=YO1A-DYO
        GO TO 200
```

```
  257 CONTINUE
      WRITE(9,259)
  259 FORMAT('ERROR:  SEARCH FOR YO GOING WRONG WAY TWICE')
      RETURN
*
  300 CONTINUE
*     CALCULATE PSI AND RUDDER
      PSISS=0.
      XRUDSS=0.
      PSI=(-FWIND+YV*YCD-YO*(YYO+YRUD*CPOS(1))+YRUD*CPOS(1)*YCSS)/
+      (YRUD*CHEAD(1)-YV*(XOD-XCD))
      XRUDDR=CPOS(1)*(YO-YCSS)+CHEAD(1)*(PSI-PSISS)+XRUDSS
      GO TO 500
  400 CONTINUE
      PN=(XOD-XCD)*(YRUD/XNRUD*XNV-YV)
      D1=(YRUD/XNRUD)*(XNWIND+XNYO*YO+YCD*XNV)
      D2=-FWIND-YYO*YO-YOC*YV
      PSI=(D1+D2)/PN
      XRUDDR=(XNV*(XOD-XCD)*PSI-XNWIND-XNYO*YO-YCD*XNV)/XNRUD
      PSISS=PSI
      XRUDSS=XRUDDR
  500 CONTINUE
      RUDBAR=XRUDDR
      HSHIP=HCOURS+(PSI/RADCON)
*
*
      IC=1
*      CALCULATE OPEN WATER RESISTANCE AND ENGINE THRUST
*
      IF((PSI/RADCON).LE.89.9)GO TO 105
      WRITE(9,101)PSI
  101 FORMAT(23H ERROR IN SSCON:  PSI =,F10.3)
      RETURN
  105 CONTINUE
      XD=XOD/COS(PSI)
*
      CALL RESIST(IN,XD,TSHIP,VSHIP,IOUT,ROW)
*
      THRUST=ROW-RWIND
      CSTVR(1)=XOD
      CSTVR(2)=XO
      CSTVR(3)=YOD
      CSTVR(4)=YO
      CSTVR(5)=YOI
      CSTVR(6)=PSID
      CSTVR(7)=HSHIP*RADCON
      CSTVR(8)=XODD
      CSTVR(9)=YODD
      CSTVR(10)=PSIDD
      CSTVR(11)=YO
      CSTVR(12)=YOD
      CSTVR(13)=YODD
      CSTVR(14)=YOI
      TIME=TSTART
      SAMP(1)=TIME
      SAMP(2)=XO
      SAMP(3)=XOD
```

```
         SAMP(4)=XODD
         SAMP(5)=YOI
         SAMP(6)=YO
         SAMP(7)=YOD
         SAMP(8)=YODD
         SAMP(9)=HSHIP
         SAMP(10)=PSID
         SAMP(11)=PSIDD
         SAMP(12)=XRUDDR/RADCON
         XRUDSS=XRUDDR
*
*
         SAMP(13)=VCURNT
         SAMP(14)=HCURNT
         YOFF=ABS(YO)
         ANG=ABS(HSHIP-HCOURS)*RADCON
         SAMP(16)=YOFF+0.5*(XLOA*SIN(ANG)+BEAM*COS(ANG))
         SAMP(17)=SAMP(16)
         SAMP(18)=SAMP(16)
         SAMP(19)=PSI
         RETURN
         END
*
         SUBROUTINE INTER(IN,IHEAD,XYZ,NPT,XYZ1,IOUT,XYZ2,XYZ3)
*
*         PURPOSE:  LINEAR INTERPOLATION XYZ(I,J) ARRAY FOR
*         VALUES OF XYZ2 AND XYZ3
*
         DIMENSION XYZ(10,3)
*
         IF(NPT.LE.25)GO TO 10
         WRITE(9,5)NPT
       5 FORMAT(27H ERROR IN INTER SUBROUTINE,,
+         23H NPT IS GREATER THAN 25,I5)
         RETURN
      10 CONTINUE
         IDPT=1
         IF(XYZ(1,1).GT.XYZ(NPT,1))IDPT=-1
         IPT=1
         IF(XYZ(1,1).GT.XYZ(NPT,1))IPT=NPT
*
         JPT=IPT+IDPT
         IF(XYZ1.GT.XYZ(IPT,1))GO TO 20
         DELTA=(XYZ(IPT,1)-XYZ1)/(XYZ(JPT,1)-XYZ(IPT,1))
         XYZ2=XYZ(IPT,2)-DELTA*(XYZ(JPT,2)-XYZ(IPT,2))
         DXYZ3=XYZ(JPT,3)-XYZ(IPT,3)
         IF(IHEAD.EQ.0)GO TO 15
         XSIGN=-1.
         IF(DXYZ3.LT.0.)XSIGN=1.
         IF(ABS(DXYZ3).GT.180.)DXYZ3=(360.-ABS(DXYZ3))*XSIGN
      15 CONTINUE
         XYZ3=XYZ(IPT,3)-DELTA*DXYZ3
         GO TO 50
*
      20 CONTINUE
         JPT=JPT-IDPT
      21 CONTINUE
```

```
*
      COMMON/CONST/RHOA,RHOW,GRAV,PI,RADCON,DISPLC,FPSMPH
      COMMON/SIM/TSTART,XSTART,XEND,TPRT,TDATA,DTSIM,DTPRT,
+      DTDATA,IANG,NANG
      COMMON/REACH/COURSE(10,3),NCOURS,EDGE(10,3),NEDGE,
+      XCURNT(10,3),NCURNT,WDEPTH,YCSS,HCSS
      COMMON/WEATHR/VCURNT,HCURNT,XWIND(10,3),NWIND
      COMMON/CONTRL/CPOS(4),CHEAD(3),XRUDSS,PSISS,RUDBAR
      COMMON/COEFM/XMCOEF(15),XUD,YVD,YV,YPSIDD,YPSID,XNPSDD,
+     XNVD,XNPSID,XNV,YRUD,XNRUD
      COMMON/SHIP/VSHIP,TSHIP,XLBP,XLWL,XLOA,BEAM,DRAFT,AREAP,
+     AREAF,DISPL,XIZ,RUDMAX
      COMMON/COEFCH/XYOP(7,11,2),XWPB(7),NYPB(7),XALPHA(11,3),
+      NWPB,NXALP
*
      DIMENSION STVR(15),STVRD(15),CSTVR(15)
*
      XOD=STVR(1)
      XO=STVR(2)
      YOD=STVR(3)
      YO=STVR(4)
      YOI=STVR(5)
      PSID=STVR(6)
      HSHIPR=STVR(7)
      HSHIP=HSHIPR/RADCON
      XODD=CSTVR(8)
      YODD=CSTVR(9)
      PSIDD=CSTVR(10)
*
      CALL RUDDER(IN,T,CSTVR,IOUT,XRUDDR)
*
      CALL INTER(IN,IANG,XCURNT,NCURNT,XO,IOUT,VCURNT,HCURNT)
*
      HCR=HCURNT*RADCON
      HSR=HCSS*RADCON
      XCD=VCURNT*(SIN(HCR)*SIN(HSR)+COS(HCR)*COS(HSR))
      YCD=VCURNT*(SIN(HCR)*COS(HSR)-COS(HCR)*SIN(HSR))
      XVEL=XOD-XCD
      YVEL=YOD-YCD
*
*      DETERMINE MANEUVERING COEFFICIENTS
*
      CALL CHCOEF(IN,YO,XO,XVEL,YVEL,IOUT,YYO,XNYO)
*
*      DETERMINE EXTERNAL FORCES AND MOVEMENTS
*
      CALL INTER(IN,IANG,COURSE,NCOURS,XO,IOUT,YCOURS,HCOURS)
*
      PSI=(HSHIP-HCOURS)*RADCON
*
      CALL WIND(IN,T,HSHIP,XOD,YOD,IOUT,RWIND,FWIND,XNWIND)
*
*      CALCULATE YODD
*
      XG=0.0
      XM=DISPLC*DISPL/GRAV
      YODD=(FWIND+YRUD*XRUDDR+YYO*YO+YV*(YOD-YCD)-YV*
```

```
+      (XOD-XCD)*PSI+(YPSID-YVD*XOD)*PSID+(YPSIDD-XM*XG)*PSIDD)/
+      (XM-YVD)
*
*       CALCULATE PSIDD
*
       PSIDD=(XNWIND+XNRUD*XRUDDR+XNYO*YO+XNV*(YOD-YCD)-
+      XNV*(XOD-XCD)*PSI+(XNPSID-XNVD*XOD)*PSID+
+      (XNVD-XM*XG)*YODD)/(XIZ-XNPSDD)
*
*       CALCULATE XODD
*
       XODD=0.
*
       STVRD(1)=XODD
       STVRD(2)=XOD
       STVRD(3)=YODD
       STVRD(4)=YOD
       STVRD(5)=YO-YCOURS
       STVRD(6)=PSIDD
       STVRD(7)=PSID
       CSTVR(8)=XODD
       CSTVR(9)=YODD
       CSTVR(10)=PSIDD
       RETURN
       END
*
       SUBROUTINE WIND(IN,T,HSHIP,XOD,YOD,IOUT,RWIND,FWIND,XNWIND)
*
*
*       PURPOSE:  TO CALCULATE FRONTAL AND LATERAL FORCES AND
*       YAW MOMENT DUE TO WIND
*
       COMMON/CONST/RHOA,RHOW,GRAV,PI,RADCON,DISPLC,FPSMPH
       COMMON/SIM/TSTART,XSTART,XEND,TPRT,TDATA,DTSIM,DTPRT,
+       DTDATA,IANG,NANG
       COMMON/REACH/COURSE(10,3),NCOURS,EDGE(10,3),NEDGE,
+       XCURNT(10,3),NCURNT,WDEPTH,YCSS,HCSS
       COMMON/WEATHR/VCURNT,HCURNT,XWIND(10,3),NWIND
       COMMON/CONTRL/CPOS(4),CHEAD(3),XRUDSS,PSISS,RUDBAR
       COMMON/COEFM/XMCOEF(15),XUD,YVD,YV,YPSIDD,YPSID,XNPSDD,
+      XNVD,XNPSID,XNV,YRUD,XNRUD
       COMMON/SHIP/VSHIP,TSHIP,XLBP,XLWL,XLOA,BEAM,DRAFT,AREAP,
+      AREAF,DISPL,XIZ,RUDMAX
       COMMON/COEFCH/XYOP(7,11,2),XWPB(7),NYPB(7),XALPHA(11,3),
+       NWPB,NXALP
*
*       DETERMINE WIND VELOCITY AND DIRECTION
*
       CALL INTER(IN,IANG,XWIND,NWIND,T,IOUT,VWIND,HWIND)
*
*       FRONTAL AND LATERAL WIND FORCES
*
       VWIND=VWIND*FPSMPH
       HSHIPR=HSHIP*RADCON
       HWIND=HWIND+180.
       IF(HWIND.GT.360.)HWIND=HWIND-360.
       HWINDR=HWIND*RADCON
```

```
      HSR=HCSS*RADCON
      VWX=VWIND*(SIN(HWINDR)*SIN(HSR)+COS(HWINDR)*COS(HSR))
      VWY=VWIND*(SIN(HWINDR)*COS(HSR)-COS(HWINDR)*SIN(HSR))
      VWX=VWX-XOD
      VWY=VWY-YOD
      HWIND=-99999.
      IF(ABS(VWX).LT.0.0001.AND.VWY.GE.0.0)HWIND=HCSS+90.
      IF(ABS(VWX).LT.0.0001.AND.VWY.LT.0.0)HWIND=HCSS+270.
      IF(VWX.GE.0.0001)HWIND=HCSS+ATAN(VWY/VWX)/RADCON
      IF(VWX.LE.-0.0001)HWIND=HCSS+180.+ATAN(VWY/VWX)/RADCON
      IF(ABS(HWIND+99999.).LE.1.0)WRITE(9,987)VWX,VWY
  987 FORMAT('ERROR:  HWIND HAS NOT BEEN CALCULATED:'/
     + 5X,'VWX =',E15.5,' VWY =',E15.5)
      IF(HWIND.LT.0.0) HWIND=HWIND+360.
      IF(HWIND.GE.360.0) HWIND=HWIND-360.
      VWIND=SQRT(VWX*VWX+VWY*VWY)
      TH=(HWIND-HCSS)*RADCON
      VWIND=VWIND/FPSMPH
      FORCE=1.3*RHOA*VWIND*VWIND*(AREAP*SIN(TH)*SIN(TH)+
     + AREAF*COS(TH)*COS(TH))
      RWIND=FORCE*COS(TH)
      FWIND=FORCE*SIN(TH)
*
*      YAWING WIND MOMENT
*
      HWIND=HWIND-180.
      IF(HWIND.LT.0.)HWIND=HWIND+360.
      HWINDR=HWIND*RADCON
      XNWIND=0.0
*
      RETURN
      END
      SUBROUTINE RUDDER(IN,TIME,CSTVR,IOUT,XRUDDR)
*
*      PURPOSE:  TO SIMULATE THE RUDDER COMMANDS OF A
*      HELMSMAN TO DATA ON THE POSITION AND HEADING OF
*      THE SHIP RELATIVE TO THE DESIRED POSITION AND
*      HEADING
*
      COMMON/CONST/RHOA,RHOW,GRAV,PI,RADCON,DISPLC,FPSMPH
      COMMON/SIM/TSTART,XSTART,XEND,TPRT,TDATA,DTSIM,DTPRT,
     + DTDATA,IANG,NANG
      COMMON/REACH/COURSE(10,3),NCOURS,EDGE(10,3),NEDGE,
     + XCURNT(10,3),NCURNT,WDEPTH,YCSS,HCSS
      COMMON/WEATHR/VCURNT,HCURNT,XWIND(10,3),NWIND
      COMMON/CONTRL/CPOS(4),CHEAD(3),XRUDSS,PSISS,RUDBAR
      COMMON/COEFM/XMCOEF(15),XUD,YVD,YV,YPSIDD,YPSID,XNPSDD,
     + XNVD,XNPSID,XNV,YRUD,XNRUD
      COMMON/SHIP/VSHIP,TSHIP,XLPP,XLWL,XLOA,BEAM,DRAFT,AREAP,
     + AREAF,DISPL,XIZ,RUDMAX
      COMMON/COEFCH/XYOP(7,11,2),XWPB(7),NYPB(7),XALPHA(11,3),
     + NWPB,NXALP
      DIMENSION CSTVR(15)
*
      IF(TIME.GT.(TDATA-0.0001))GO TO 100
      XRUDDR=RUDBAR
      CSTVR(15)=XRUDDR
```

```
      RETURN
  100 CONTINUE
      XO=CSTVR(2)
*
      CALL INTER(IN,IANG,COURSE,NCOURS,XO,IOUT,YCOURS,HCOURS)
*
      YO=CSTVR(11)
      YOD=CSTVR(12)
      YODD=CSTVR(13)
      YOI=CSTVR(14)
      HSHIPR=CSTVR(7)
      HSHIP=HSHIPR/RADCON
      PSID=CSTVR(6)
      PSIDD=CSTVR(10)
*
*      DETERMINE NEW XRUDDR SETTING
*
      ANGLER=(HSHIP-HCOURS)*RADCON-PSISS
      XRUDDR=CPOS(1)*(YO-YCOURS)+CPOS(2)*YOD+CPOS(3)*YODD+
+      CPOS(4)*YOI+CHEAD(1)*(ANGLER)+CHEAD(2)*PSID+CHEAD(3)*PSIDD+
+      XRUDSS
      IF(ABS(XRUDDR).GT.RUDMAX)XRUDDR=RUDMAX*ABS(XRUDDR)/XRUDDR
      CSTVR(15)=XRUDDR
      RUDBAR=XRUDDR
      TDATA=TDATA+DTDATA
      RETURN
      END
      SUBROUTINE PRT(IN,SAMP)
      DIMENSION SAMP(19)
*
      WRITE(1,20)SAMP(1),SAMP(2),SAMP(6),SAMP(9),SAMP(3),SAMP(7),
+      SAMP(10),SAMP(12),SAMP(16)
   20 FORMAT(F8.2,2F9.2,F8.2,4F9.2,F10.2)
      RETURN
      END
*
      SUBROUTINE CHCOEF(IN,YO,XO,XVEL,YVEL,IOUT,YYO,XNYO)
      COMMON/CONST/RHOA,RHOW,GRAV,PI,RADCON,DISPLC,FPSMPH
      COMMON/SIM/TSTART,XSTART,XEND,TPRI,TDATA,DTSIM,DTPRT,
+      DTDATA,IANG,NANG
      COMMON/REACH/COURSE(10,3),NCOURS,EDGE(10,3),NEDGE,
+      XCURNT(10,3),NCURNT,WDEPTH,YCSS,HCSS
      COMMON/WEATHR/VCURNT,HCURNT,XWIND(10,3),NWIND
      COMMON/CONTRL/CPOS(4),CHEAD(3),XRUDSS,PSISS,RUDBAR
      COMMON/COEFM/XMCOEF(15),XUD,YVD,YV,YPSIDD,YPSID,XNPSDD,
+     XNVD,XNPSID,XNV,YRUD,XNRUD
      COMMON/SHIP/VSHIP,TSHIP,XLEP,XLWL,XLOA,BEAM,DRAFT,AREAP,
+     AREAF,DISPL,XIZ,RUDMAX
      COMMON/COEFCH/XYOP(7,11,2),XWPB(7),NYPB(7),XALPHA(11,3),
+      NWPB,NXALP
*
      VEL=SQRT(XVEL*XVEL+YVEL*YVEL)
*
      CALL INTER(IN,NANG,EDGE,NEDGE,XO,IOUT,CE1,CE2)
*
      WPB=(ABS(CE1-CE2))/BEAM
      HPT=WDEPTH/DRAFT
```

```
*
      IF(ABS(YO).GE.0.001)GO TO 10
      YYO=0.
      XNYO=0.
      RETURN
*
   10 CONTINUE
      DO 40 I=1,NWPB
      IF(WPB.GT.XWPB(I))GO TO 30
      IF(I.NE.1)GO TO 20
      WRITE(9,15)
   15 FORMAT(' WPB IS LESS THAN XWPB(1)')
      YYO=0.
      XNYO=0.
      RETURN
*
   20 CONTINUE
      AYOPB=ABS(YO)/BEAM
      IF(AYOPB.GT.0.0)GO TO 25
      YYO=0.
      XNYO=0.

      RETURN
   25 CONTINUE
      I1=I-1
      GO TO 45
   30 CONTINUE
      IF(I.LT.NWPB)GO TO 40
      YYO=0.
      XNYO=0.
      RETURN
   40 CONTINUE
      WRITE(9,42)
   42 FORMAT('ERROR IN SUBROUTINE CHCOEF AT STATEMENT 40')
      RETURN
   45 CONTINUE
      DO 200 II=1,2
      IF(II.EQ.1)L=I1
      IF(II.EQ.2)L=I
      MYPB=NYPB(L)
      DO 90 J=1,MYPB
      IF(AYOPB.GT.XYOP(L,J,1))GO TO 80
      IF(J.EQ.1)GO TO 50
      GO TO 70
*
   50 CONTINUE
*       AYOPB CAN NOT BE LESS THAN ZERO, VALUE OF XYOP(L,1,1)
      WRITE(9,60)AYOPB,XYOP(L,9,1)
   60 FORMAT('ERROR IN SUBROUTINE CHCOEF AT STATEMENT 50.'/
     +  'AYOPB OF ',F10.2,' LESS THAN ',F10.2)
      RETURN
   70 CONTINUE
*       AYOPB BETWEEN J AND J-1
      DTAB1=XYOP(L,J,1)-XYOP(L,J-1,1)
      DTAB2=XYOP(L,J,2)-XYOP(L,J-1,2)
      DYOPB=AYOPB-XYOP(L,J-1,1)
      YOPP=XYOP(L,J-1,2)+DTAB2*DYOPB/DTAB1
```

```
      GO TO 100
*
   80 CONTINUE
      IF(J.LT.NYPB(L))GO TO 90
*       AYOPB GREATER THAN FINAL VALUE OF XYOP
      DTAB1=XYOP(L,J,1)-XYOP(L,J-1,1)
      DTAB2=XYOP(L,J,2)-XYOP(L,J-1,2)
      DYOPB=AYOPB-XYOP(L,J,1)
      YOPP=XYOP(L,J,2)+DTAB2*DYOPB/DTAB1
      GO TO 100
   90 CONTINUE
*      SHOULD NOT EXIT DO LOOP HERE
      WRITE(9,95)
   95 FORMAT('ERROR IS SUBROUTINE CHCOEF AT STATEMENT 90')
      RETURN
  100 CONTINUE
*
      IF(II.EQ.1)YOP1=YOPP
      IF(II.EQ.2)YOP2=YOPP
  200 CONTINUE
*      VALUE OF YOP BETWEEN YOP1 AND YOP2
      DWPB1=XWPB(I)-XWPB(I1)
      DWPB2=WPB-XWPB(I1)
      DYOP=YOP2-YOP1
      YOP=YOP1+DYOP*DWPB2/DWPB1
*
      DO 300 K=1,NXALP
      IF(HPT.GT.XALPHA(K,1))GO TO 180
      IF(K.NE.1)GO TO 170
*       HPT LESS THAN FIRST VALUE OF XALPHA
      DXAL1=XALPHA(2,1)-XALPHA(1,1)
      DXAL2=XALPHA(2,2)-XALPHA(1,2)
      DXAL3=XALPHA(2,3)-XALPHA(1,3)
      DHPT=XALPHA(1,1)-HPT
      XBAR=XALPHA(1,2)-DXAL2*DHPT/DXAL1
      ALPHA=XALPHA(1,3)-DXAL3*DHPT/DXAL1
      GO TO 220
*
  170 CONTINUE
*      HPT BETWEEN K-1 AND K
      DXAL1=XALPHA(K,1)-XALPHA(K-1,1)
      DXAL2=XALPHA(K,2)-XALPHA(K-1,2)
      DXAL3=XALPHA(K,3)-XALPHA(K-1,3)
      DHPT=HPT-XALPHA(K-1,1)
      XBAR=XALPHA(K-1,2)+DXAL2*DHPT/DXAL1
      ALPHA=XALPHA(K-1,3)+DXAL3*DHPT/DXAL1
      GO TO 220
*
  180 CONTINUE
      IF(K.NE.NXALP)GO TO 300
*      HPT ABOVE FINAL VALUE OF XALPHA
      DXAL1=XALPHA(K,1)-XALPHA(K-1,1)
      DXAL2=XALPHA(K,2)-XALPHA(K-1,2)
      DXAL3=XALPHA(K,3)-XALPHA(K-1,3)
      DHPT=HPT-XALPHA(K,1)
      XBAR=XALPHA(K,2)+DXAL2*DHPT/DXAL1
      ALPHA=XALPHA(K,3)+DXAL3*DHPT/DXAL1
```

```
      GO TO 220
  300 CONTINUE
*      SHOULD NOT EXIT DO LOOP HERE
      WRITE(9,205)
  205 FORMAT('ERROR IN SUBROUTINE CHCOEF AT STATEMENT 300')
      RETURN
*
  220 CONTINUE
      YYO=0.5*RHOW*XLWL*DRAFT*VEL*VEL*YOP*ALPHA/ABS(YO)
      XNYO=XBAR*XLWL*YYO
      RETURN
      END
```

230 Copies

U.S. DEPARTMENT OF TRANSPORTATION
RESEARCH AND SPECIAL PROGRAMS ADMINISTRATION
TRANSPORTATION SYSTEMS CENTER
KENDALL SQUARE, CAMBRIDGE, MA. 02142

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF TRANSPORTATION
513

U.S. MAIL