REPORT NO. CG-D-42-74

VESSEL SAFETY MODEL

VOLUME III
PROGRAMMERS' MANUAL

D. Kahn
T. Talbot
J. Woodard

JANUARY 1974
FINAL REPORT

Prepared for
DEPARTMENT OF TRANSPORTATION
UNITED STATES COAST GUARD
Office of Research and Development
Washington DC 20591

Technical Report Documentation Page

1. Report No. CG-D-42-74	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle VESSEL SAFETY MODEL Volume III - Programmers' Manual		5. Report Date January 1974
		6. Performing Organization Code
7. Author(s) D. Kahn, T. Talbot, J. Woodard		8. Performing Organization Report No. DOT-TSC-USCG-72-5-III
9. Performing Organization Name and Address Department of Transportation Transportation Systems Center Kendall Square Cambridge MA 02142		10. Work Unit No. (TRAIS) CG401-R4005
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address Department of Transportation United States Coast Guard Office of Research & Development Washington DC 20591		13. Type of Report and Period Covered Final Report July 1971 - June 1972
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A computer model which mathematically simulates the ship's movement in defined waterways is described. Volume I presents the capabilities and usefulness for ship traffic lane selection, development of the equations of motion, and the collision and grounding probability analysis used in the computer program. Volume II of the report consists of a complete Users' Manual. Volume III is a self-contained Programmers' Manual.

17. Key Words	18. Distribution Statement
·Harbor ·Probability (Collisions/Groundings)	DOCUMENT IS AVAILABLE TO THE PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22151.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	148	

Form DOT F 1700.7 (8–72)

PREFACE

This volume presents and explains the individual programs in the mathematical model. Familiarization with the details of these programs will provide the programmer with the necessary information to effect program modifications where needed. As for Volumes II and III, the authors are happy to acknowledge the extensive effort of Mr. Leon Tritter in the preparation of this document.

CONTENTS

CONTENTS (CONT'D)

LIST OF TABLES

1. INTRODUCTION

1.1 MODEL OVERVIEW

The vessel safety model provides simulated movement of ships through a defined waterway, and also furnishes information on the probabilities of collision and grounding. The ships are moved through the waterway in accordance with programmed ship motion equations, and the probabilities are printed out according to derived equations for the respective probabilities. The model is capable of a further printed output which indicates the closest point of approach (CPA) between any pair of ships in the waterway. Supplementary information is available in Section 3 of Volume I with respect to input data requirements, the background and development of the vessel safety model, the programs and sub-programs associated with the model, and a flow chart which shows the interrelationship between the programs, subprograms, and functions in the model operation.

1.2 KEY POINTS OF PROGRAM DESIGN

Two features stand out as key factors in the overall design of the model:

a. Command stack

b. Variability in time steps.

1.2.1 Command Stack

The command stack provides a set of commands as input to the model. The command stack concept permits the performing of modifications, typical of which are the evasive maneuvers during execution of a run.

1.2.2 Variability in Time Steps

This feature permits variations in time steps to be performed within a single run. By means of this capability the model operates with greater accuracy and greater economy, both as the

2. MAIN PROGRAM HAR

2.1 DESCRIPTION

The following description covers the logic used in the main program HAR. A flow chart of the logic of the complete program is provided in Figure 2-1, and time variables associated with the program are indicated in Figure 2-2. The main variables used in this program are listed and defined in Table 2-1, and the complete details are supplied in the program listing of Table 2-2.

a. The model uses ship commands (or maneuvers) as track information. The commands, with their command number or mode, are defined as follows:

COMMAND	MODE	VALUE
Constant path	1	Straight ahead for (value) ft
Acceleration	2	Accelerate to a speed of (value) ft/sec
Deceleration	3	Decelerate to a speed of (value) ft/sec
Left turn	4	Turn left (value) rad
Right turn	5	Turn right (value) rad

(The units are those used during execution of the model, and are changed for printout.) Each ship has a set of commands stored in core which are executed sequentially. Storing the commands in core is advantageous in that it provides ease of modification for use in decision routines. Thus, if a grounding hazard is detected, a turn command may be inserted into the command stack and executed next. The turn command is then followed by the remaining commands, or by new commands, to attempt to get the ship back on its correct course.

b. The initial commands are input by subroutines TRKRD and CMDRD. For this model only the initial commands are used. For the

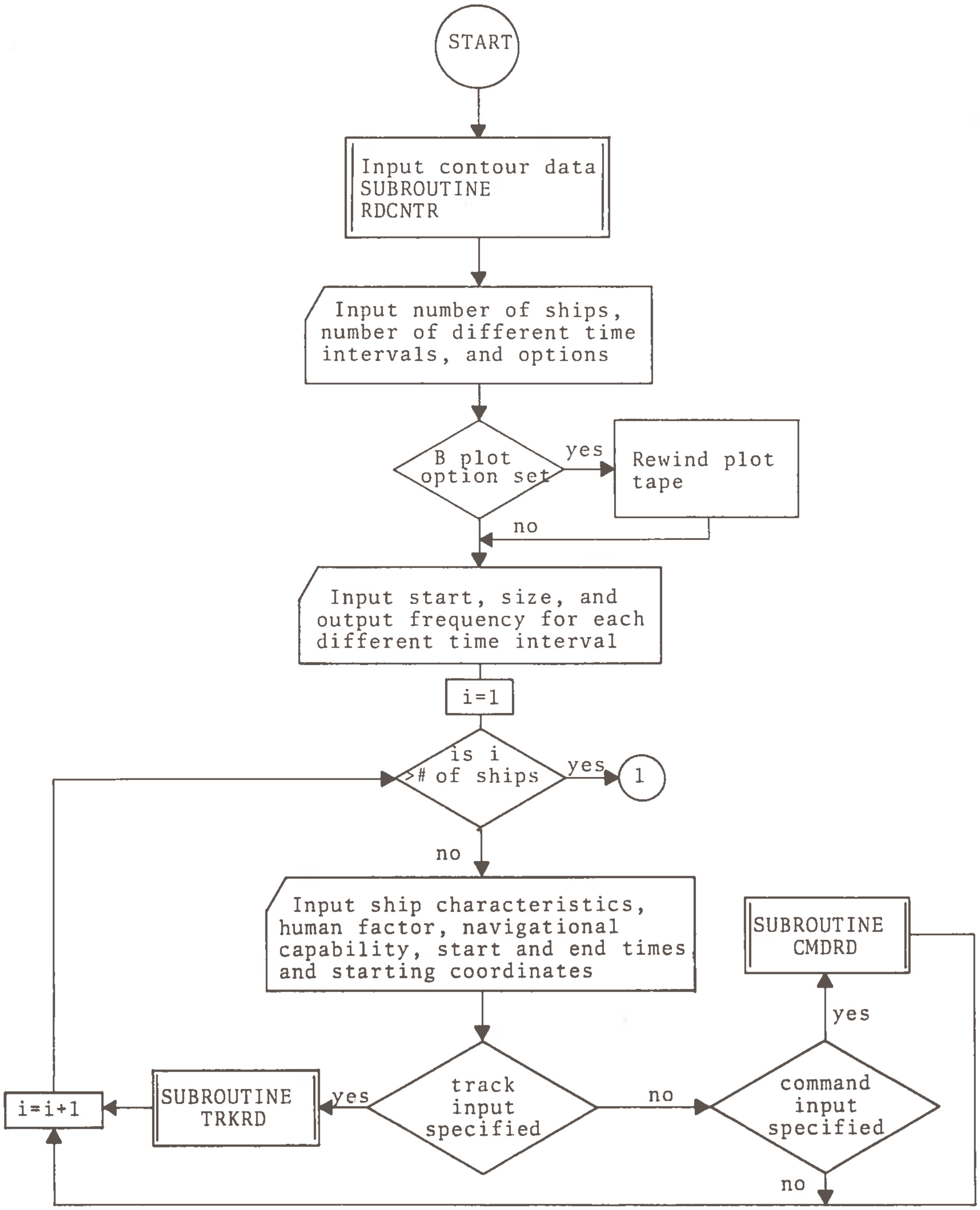

Figure 2-1. Flow Chart of Logic for Main Program HAR

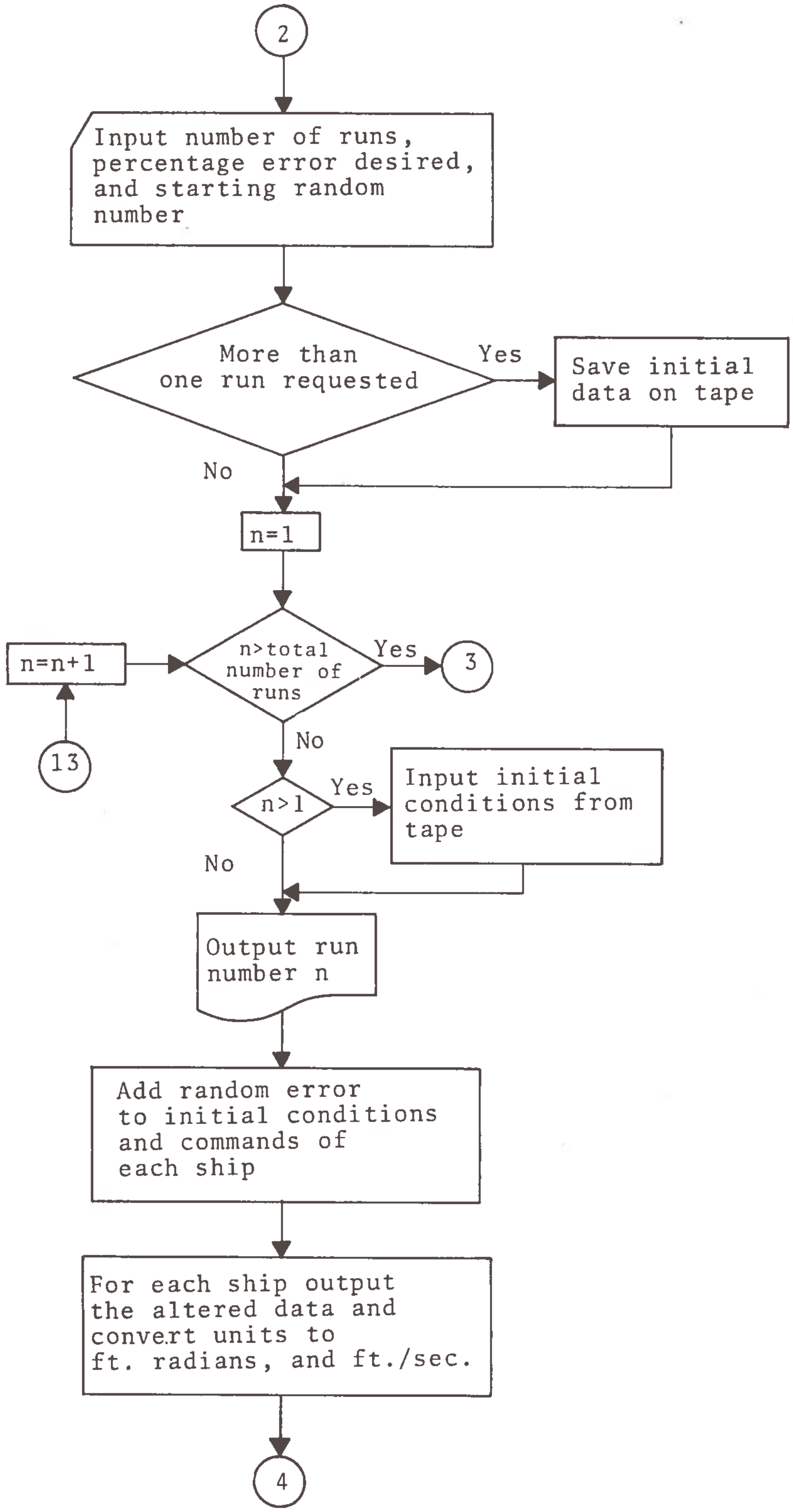

Figure 2-1. Flow Chart of Logic for Main Program HAR (Cont'd)

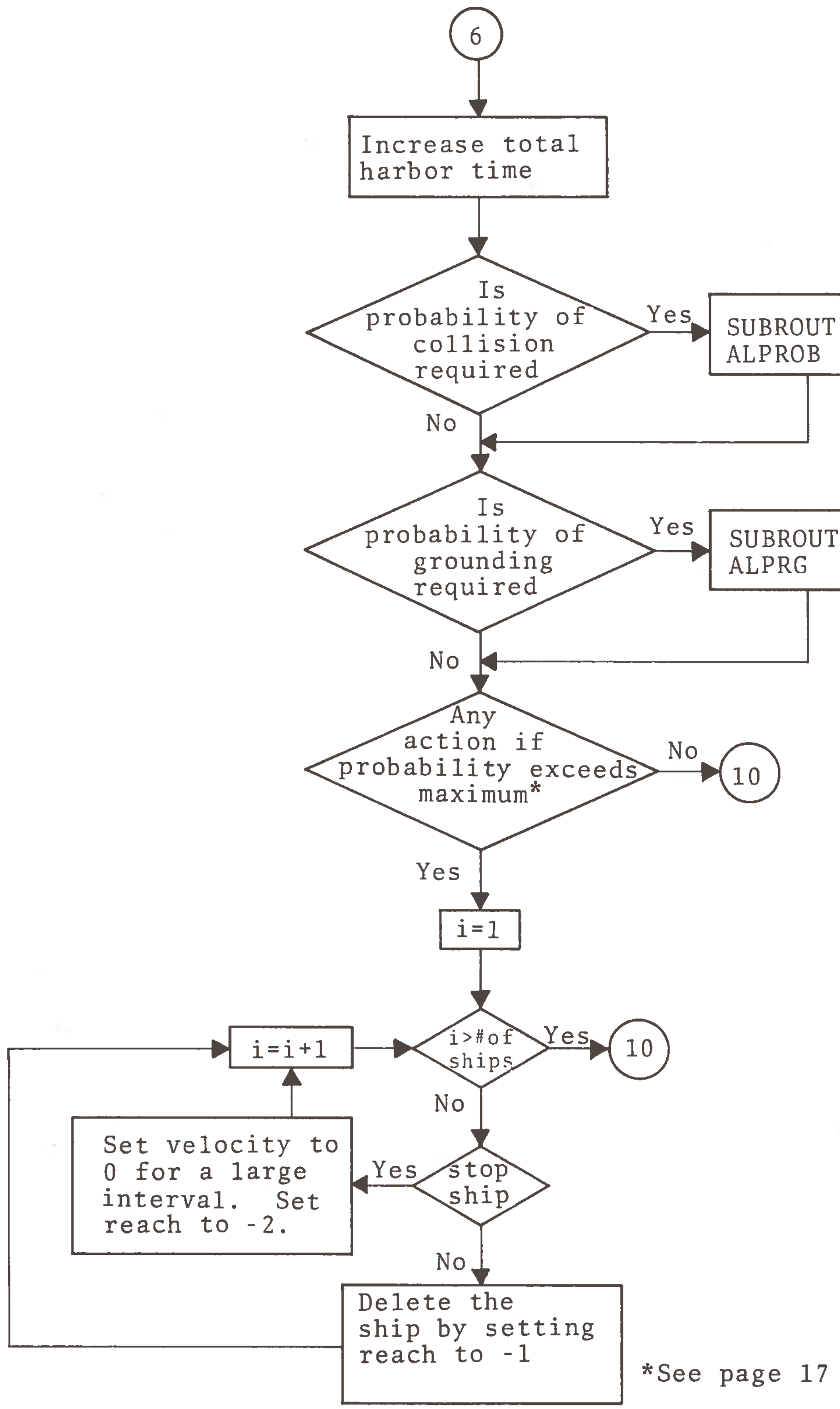

Figure 2-1. Flow Chart of Logic for Main Program HAR (Cont'd)

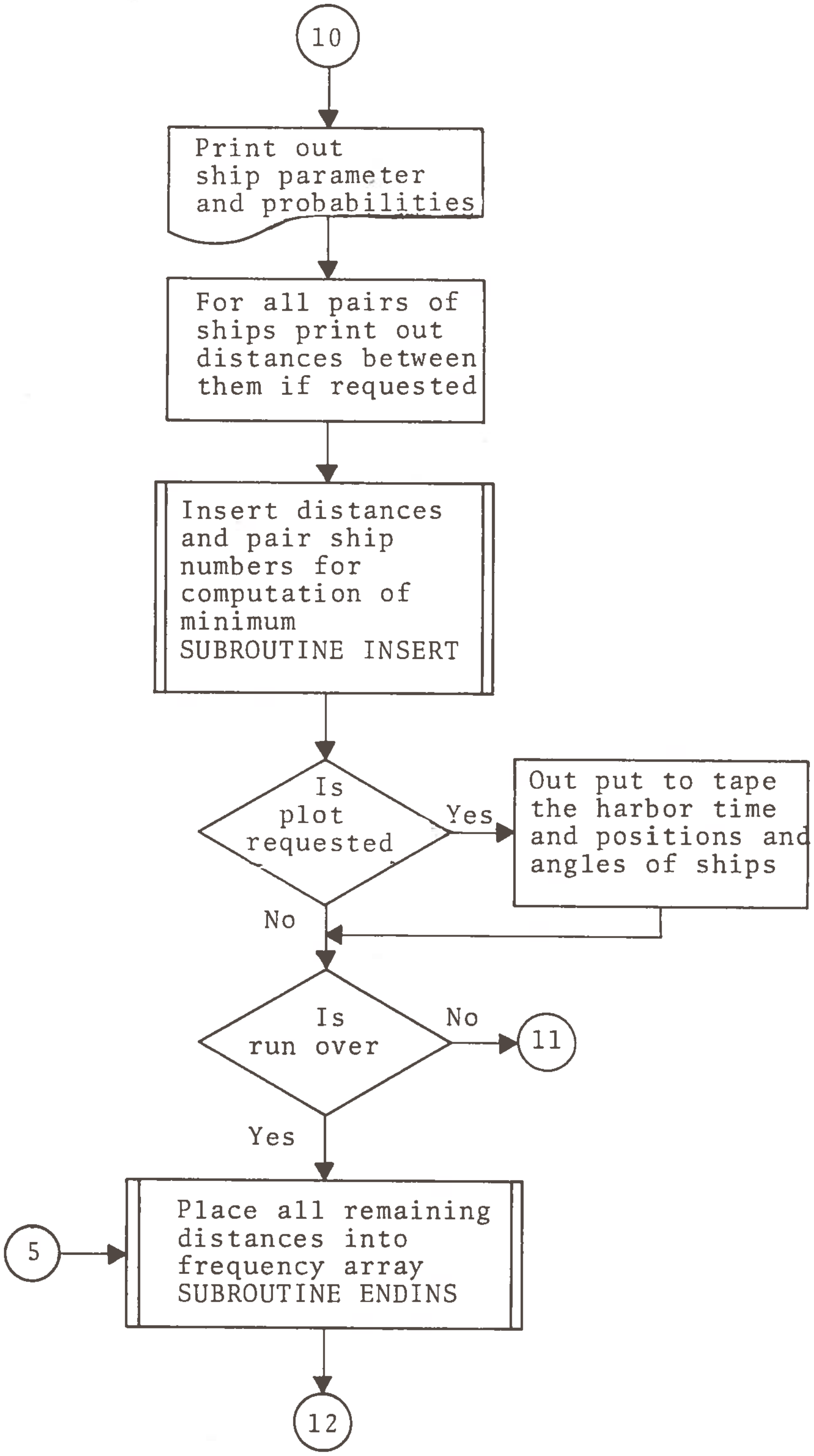

Figure 2-1. Flow Chart of Logic for Main Program HAR (Cont'd)

Figure 2-2 indicates the time variables relating the parameters TLSEG, TIMINT, and TLINT (defined in Table 2-1).

d. The remainder of the main program contains the following:

(1) Input/Output

(2) Probabilities of collision and grounding

(3) Closest point of approach (CPA) calculation for every pair of ships

(4) Frequency distribution of CPAs

(5) Option for using different time intervals during the run

(6) Option for multiple runs, changing some input data randomly

(7) Option for generating a ship's position tape to be used for plotting

(8) An option for deleting or stopping a ship (or pair of ships) when the probabilities are greater than a specified amount

(9) An option for printing either all distances between ships or just the distances less than a specified amount.

These are discussed as they occur in the main program, which is covered in the order of the listing (provided in Table 2-2).

e. The dimension statements limit the number of ships, the number of commands in the command stack, the number of different time intervals, and so on. Care must be exercised not to exceed these dimensions. The most delicate dimension is the number of commands in the command stack when commands are to be inserted by the user through some decision routine. The dimensions may be increased, depending on the amount of core storage available.

f. The arrays HUFA and NAVC are used to input values only, and the arrays IWRK, JWRK to output values only. Each pair is not needed for further computation; therefore, each pair is equivalenced to save storage. The arrays containing characters AMODE, TITLE, PCOL, and PGRN are stored with four characters per word for compatibility with the IBM 360. Most other machines hold more than four characters per word, and will cause no conversion problems if the program needs conversion to a particular machine. (The model was run on the IBM 7094 and the CDC 6600 with no conversion excepting control cards.) The data for the matrix AMODE are stored (by the compiler) columnwise (i.e., completing the left index first).

g. The subroutine RDCNTR is called to read in the harbor depth contour data required by the probability of grounding calculations. All data inputs are by cards, and the input unit is the standard unit 5. The first READ statement reads in the options desired and some parameters. The options are straightforward, and are described in the listing. NTIMS, the number of different time intervals, is used to allow variation in the time step used by the model. A maximum of ten is set by the dimensions of the variables TIMES, TIMINT, and PRTFRE. For each different interval size (TIMINT) its duration is also required. The variable TIMES contains the end time in seconds (relative to zero) for each interval. The variable PRTFRE is used to define the printout frequency. If only the motion equations were involved, there would be no use for PRTFRE, but some calculations are dependent on the time interval size for accuracy (e.g., probability of collision, probability of grounding, and CPA calculations). If PRTFRE = n, status printout is generated every $n^{\underline{th}}$ time interval.

h. The DO 5 loop reads in for each ship its name, characteristics, on-board navigational capability, start and end times, starting position, and maneuvers. Each ship in the program during printout is identified by a number rather than the name of the ship. The numbers, initially, are printed along with the corresponding name for reference. The names are then no longer used, and the number assigned is just the current value of the DO loop index. The twenty characteristics (weight, draft, and so on) are obtained from program CHAR. The variable NAVC is the position error (in meters) due to on-board navigational equipment inaccuracies,

l. IX is to be used as input to the subroutine RANDU, which requires as input a random integer, and which returns a random number and another random integer to be used as input to the next CALL to RANDU.

m. PMAX is the probability value used with the input variable POPT. Probabilities are stored in the arrays ALP and ALPG. The array ALPG contains the log of the probability of no-grounding, and ALP contains the log of the probability of no collision. The option POPT will require a check of probabilities against the value PMAX. The IF statement would be of the form (using collision as an example):

IF (PROB(collision) > PMAX) Go to XX

PROB(collision) is equal to 1 - exp(ALP)

Therefore $1 - \exp(ALP) > PMAX$

$$- \exp(ALP) > PMAX - 1$$

$$ALP < Ln(1 - PMAX)$$

By setting PMAX = ALOG(1 - PMAX), PMAX is now comparable to ALP (or ALPG) by using IF(PMAX > ALP) Go to XX.

n. Next, if the run number IRUN is greater than 1, the initial values are read from the scratch device.

o. Some initialization is now done. NUM and OCCUR will be explained under subroutine INSERT (Section 10). The initial values are also modified here and printed. The percentage used to change the value of a command depends on the mode of the command. Mode 1 is a distance, modes 2 and 3 are velocities, and modes 4 and 5 are angles. The units are then converted to those used by the motion equations.

p. The DO 616 loop obtains the first command and the time required to execute the command for each ship. The sine and cosine of the heading of the ships are saved to prevent them from being computed every time step, since the headings will be changed only during turns. The ship commands that will be used for execution are then printed.

by checking the remainder after division (the MOD function) against 0]. If the remainder is not equal to zero, ICHECK will be false, and all printout will be bypassed by jumping to statement 540. Otherwise, the following printout is generated. The title is first written. (This depends on PCALC.) The title for every page used during the execution loop is stored in the array TITLE, and to this is added, depending on PCALC, a probability part. The DO 350 loop prints the ship number, the time, the ship's position, velocity, heading, current maneuver, and the probabilities that were requested. ITI is used as the correct column member for the printing of the maneuver stored in the matrix AMODE. PC and PG are the cumulative probabilities of collision and grounding, respectively. The array ALP contains the log of the probability of no collision; therefore, 1 - exp(ALP) is the probability of collision. The same is true of ALPG for grounding. Distances are next computed between all ships not considered out. Ship 1 is checked against ships 2 through NSHIPS, ship 2 is checked against ships 3 through NSHIPS, and so on until all the distances are computed. The variable DLTR is set true if the computed distance is less than the input value RADI, and is false otherwise. DLTR is used in subroutine INSERT for a CPA check. Subroutine INSERT decides if the current distance (if less than RADI) is the minimum between two ships and not just a distance in the interval 0 to RADI. The print option, PRTDIS = 1 results in printing all distances less than RADI. Therefore, if PRTDIS = 1 and the distance is not less than RADI, this distance is not to be printed, and control is sent to the end of the loop for the next distance calculation. Five distances are printed per line, and counter NM is used to check if five distances have been accumulated. Along with each distance printed the two ship numbers I and J are also printed, and an asterisk or no asterisk (blank) is printed next to the distances, depending on whether or not it is less than RADI. These also must be saved in sets of five. CHAR will contain the asterisk or blank, WORK will contain the distance, and the two arrays IWRK and JWRK will contain the ship numbers. When NM reaches five the line is printed, and NM is reset to zero for the next line. After the distance

TABLE 2-1. LIST OF VARIABLES FOR MAIN PROGRAM HAR

ALPG	Array containing the log of the cumulative probability of no-grounding
AMODE	Array containing the characters used in the printout of the maneuvers
ALP	Array containing the log of the cumulative probability of no-collision
ANGLE	Variable containing the heading of a ship in degrees (used during printout)
AMETER	Conversion factor for changing meters to feet
ADV	Array containing the length of the advance for each ship
ANG	Array containing the heading of each ship in radians
BLANK	Variable containing a Hollerith blank
CHAR	Array containing the character BLANK - If distance between the two ships being checked is greater than RADI '*' - If distance is less than RADI
COSANG	Array containing the cosines of the headings of each ship
DLTR	Logical variable set true if distance between ships being checked is less than RADI, otherwise false
DX	x - component of the distance between two ships
DY	y - component of the distance between two ships
DOT	Variable containing the Hollerith character 'o'.

TABLE 2-1. LIST OF VARIABLES FOR MAIN PROGRAM HAR (CONT'D)

MODE	Array containing the current maneuver of each ship: = 1, straight path (constant velocity) = 2, accelerating = 3, decelerating = 4, turning left = 5, turning right
MODCD	Array containing the maneuver for each command in the command stack
MODEIN	Variable signifying means of input of ship maneuvers: = 1, track parameters will be input, and the commands for the command stack will be computed. = 2, the commands will be directly input and inserted into the command stack.
NAME	Array containing the name of the ship as input. It is used only for printout in the name-number correspondence; thereafter, only the number is used.
NAV	Total on-board navigational capability. The product of the human factor and the on-board equipment accuracy (distance units). It is combined with the harbor navigational aids accuracy to form the position ellipse.
NAVC, HUFA	Variables used in the computation of total on-board accuracy. NAVC is the equipment accuracy (maximum distance error from correct position) and HUFA is the human factor.
NCMDS	The total number of commands in the command stack for each ship.

TABLE 2-1. LIST OF VARIABLES FOR MAIN PROGRAM HAR (CONT'D)

PG	Probability of grounding (cumulative)
PGRN	Array containing the title for the probability of grounding printout
PHASE	Array containing the phase of the command being executed for each ship. Modes 3, 4, and 5 have two phases. Mode 3 (deceleration) Phase 1 - time for reversing engines Phase 2 - deceleration Mode 4 (left turn) Phase 1 - time for rudder response Phase 2 - turn Mode 5 (right turn) Same as mode 4
PMAX	Probability used in conjunction with POPT
POPT	Variable used to specify action to be taken if probability of collision (or grounding) exceeds PMAX: = 1, take no action. = 2, stop ships involved. = 3, delete ships involved.
POSX	x-coordinate of a ship before it is moved
POSY	y-coordinate of a ship before it is moved
PROB	Value of the probability to be used in the POPT option
PRTCNT	The number of times through the execution loop. If it is a multiple of PRTFRE, output is generated.

TABLE 2-1. LIST OF VARIABLES FOR MAIN PROGRAM HAR (CONT'D)

	SC_{10} - SC_{20} Calculated in Program
SC_{10}	$SC_{10} = 429\ SC_5/SC_4(1.6889)$ (lb)
SC_{11}	$SC_{11} = SC_{15}\ (VMAX)^2/SC_{10}$ (ft)
VMAX	$VMAX = 1.6889\ SC_4$ (ft/sec)
SC_{12}	$SC_{12} = (1.56)\ SC_5/SC_6(VMAX)^2$ $(sec/ft)^2$
SC_{13}	$SC_{13} = \sqrt{SC_{12}}$ (sec/ft)
SC_{14}	$SC_{14} = SC_{15}/(0.95)\ SC_{10}(sec^2/ft)$
SC_{15}	$SC_{15} = (1.05)\ (2240).\ SC_7/g$ $(lb\text{-}sec^2/ft)$
SC_{16}	$SC_{16} = (0.016)SC_1 . SC_3 . \sin$ $(RA \cdot RDN)$ $[lb/(ft/sec)^2]$
SC_{17}	$SC_{17} = (VMAX)^2/(0.025)\ (RA)$ (RDN) (ft^2/sec)
SC_{18}	Ship's turning velocity variable (ft/sec)
SC_{19}	Ship's turning velocity variable (dimensionless)
SC_{20}	Ship's turning velocity variable $(ft/sec)^{-1}$ Note: RA = Ship's maximum rudder angle (degrees) RDN = Conversion from degrees to radians (0.01745329252 rad/deg)
SHIP	Integer specifying the current ship number being processed
SHR	The ship reach, computed in subroutine GETTIM. It is set equal to -1 if a ship is permanently deleted, and to -2 if a ship has been given zero velocity in the POPT option.

TABLE 2-1. LIST OF VARIABLES FOR MAIN PROGRAM HAR (CONT'D)

VEL	The current velocity of each ship in ft/sec
VT	The turning velocity (ft/sec) of a ship, if turning
YFL	Random number between 0 and 1, returned by subroutine RANDU.

TABLE 2-2. PROGRAM LISTING FOR MAIN PROGRAM HAR (CONT'D)

```
C        READ IN -
C          NSHIPS = THE TOTAL NUMBER OF SHIPS
C          NTIMS = NUMBER OF DIFFERENT TIME INTERVALS(TAUS)
C          IPLT =  1 , GENERATE A TAPE TO BE USED FOR PLOTTING
C          NOT= 1 , NO TAPE IS TO BE GENERATED
C          POPT = 1 , IF PROBABILITIES GREATER THAN PMAX,TAKE NO ACTION
C               = 2 , STOP SHIPS
C               = 3 , DELETE SHIPS
C          PCALC  = PROBABILITIES TO BE CALCULATED
C                 = 1 , NONE
C                 = 2 , COLLISION ONLY
C                 = 3 , GROUNDING ONLY
C                 = 4 , COLLISION AND GROUNDING
C          PMAX = PROBABILITY USED WITH POPT
C          RADI = MAXIMUM DISTANCE APART SHIPS WILL BE CHECKED
C                 FOR CLOSEST POINT OF APPROACH
C          PRIDIS = 1 , PRINT DISTANCES BETWEEN SHIPS THAT ARE LESS
C                       THAN RADI
C          NOT= 1 , PRINT ALL DISTANCES BETWEEN SHIPS
C          NINT = NUMBER OF INTERVALS BETWEEN 0 AND RADI FOR
C                   FREQ. PLOT
C
      READ(5,20)NSHIPS,NTIMS,IPLT,POPT,PCALC,PMAX,RADIM,PRTDIS,NINT
   20 FORMAT(5I5,2F15.5,2I5)
      RADI=RADIM*AMETER
      IF(IPLT.EQ.1)REWIND 9
      READ(5,26)(TIMES(I),TIMINT(I),PRTFRE(I),I=1,NTIMS)
   26 FORMAT(2F15.5,I5)
      WRITE(6,910)
  910 FORMAT(1H1,20X,8HSHIP.NO.,20X,4HNAME)
      DO 5 I=1,NSHIPS
      READ(5,915)NAME
      WRITE(6,940)I,NAME
  915 FORMAT(5A4)
  400 FORMAT(I5,5F13.3,3X,3A4,A1,4X,2E15.6)
  940 FORMAT(20X,I5,15X,5A4)
C
C          READ IN THE TWENTY SHIP CHARACTERISTICS
C
      READ(5,25)(SC(J,I),J=1,20)
C
C        READ IN THE HUMAN FACTOR AND THE ON-BOARD NAV.
C          CAPABILITY(METERS)
C
      READ(5,25)HUFA(I),NAVC(I)
C
C        COMPUTE THE TOTAL ON-BOARD NAV. CAPABILITY
C
      NAV(I)=HUFA(I)*NAVC(I)
C
C        READ IN THE START AND END TIMES(SECS),AND THE STARTING
C        COORDINATES(METERS). X IS DUE NORTH,Y IS EAST.
C
      READ(5,25)START(I),ENDT(I),PX(I),PY(I)
C

C        READ THE INPUT MODE FOR SHIP MOVEMENT
```

TABLE 2-2. PROGRAM LISTING FOR MAIN PROGRAM HAR (CONT'D)

```
      N=NCMDS(SHIP)
      WRITE(6,20) SHIP
      DO 517 J=1,N
C
C        MODCD - CONTAINS THE MODE OF EACH COMMAND
C          = 1 , STRAIGHT PATH(CONSTANT VELOCITY)
C          = 2 , ACCELERATE
C          = 3 , DECELERATE
C          = 4 , LEFT TURN
C          = 5 , RIGHT TURN
C          MODES 3,4,AND 5 HAVE PHASES,NAMELY
C             MODE 3,PHASE 1 -TIME FOR REVERSING ENGINES
C                    PHASE 2 -DECELERATION
C             MODE 4,PHASE 1 - TIME FOR RUDDER TO RESPOND(ADVANCE)
C                   PHASE 2 - TURN
C             MODE 5,PHASE 1 - ADVANCE
C                     PHASE 2 - TURN
C
      L=MODCD(J,SHIP)
      GO TO(518,519,520,521,522),L
C
C        VALC - CONTAINS THE VALUE ASSOCIATED WITH EACH COMMAND
C
  518 WRITE(6,523)VALC(J,SHIP)
  523 FORMAT(10X,18HSTRAIGHT AHEAD FOR,F10.2,7H METERS)
      GO TO 517
  519 WRITE(6,524)VALC(J,SHIP)
  524 FORMAT(10X,24HACCELERATE TO A SPEED OF,F10.2,6H KNOTS)
      GO TO 517
  520 WRITE(6,525)VALC(J,SHIP)
  525 FORMAT(10X,24HDECELERATE TO A SPEED OF,F10.2,6H KNOTS)
      GO TO 517
  521 WRITE(6,526)VALC(J,SHIP)
  526 FORMAT(10X,9HTURN LEFT,F10.2,8H DEGREES)
      GO TO 517
  522 WRITE(6,527)VALC(J,SHIP)
  527 FORMAT(10X,10HTURN RIGHT,F10.2,8H DEGREES)
  517 CONTINUE
  516 CONTINUE
C
C        READ IN THE TOTAL NUMBER OF RUNS(NRUNS) AND
C          VPOS - MAXIMUM ERROR IN STARTING POSITION(PER CENT)
C          VVEL - MAXIMUM ERROR IN VELOCITY(PER CENT)
C          VANG - MAXIMUM ERROR IN HEADINGS(PER CENT)
C          VDIS - MAXIMUM ERROR IN DISTANCES(PER CENT)
C          VNAV - MAXIMUM ERROR IN ON-BOARD NAV. CAP.(PER CENT)
C          ISEED - STARTING RANDOM NUMBER
C
      READ(5,40)NRUNS,VPOS,VVEL,VANG,VDIS,VNAV ,ISEED
   40 FORMAT(I5,5F10.2,I10)
      IF(NRUNS.LT.2)GO TO 570
C
C        IF MORE THAN ONE RUN, STORE DATA NEEDED FOR EACH RUN
C
      REWIND 10
```

TABLE 2-2. PROGRAM LISTING FOR MAIN PROGRAM HAR (CONT'D)

```
      IX=IY
      PY(I)=PY(I)+(YFL*2.-1.)*VPOS/100.*PY(I)
C
C        CHANGE NAVIGATIONAL CAPABILITY
C
      CALL RANDU(IX,IY,YFL)
      IX=IY
      NAV(I)=FLOAT(NAV(I))+(YFL*2.-1.)*VNAV/100.*FLOAT(NAV(I))
      CALL RANDU(IX,IY,YFL)
      IX=IY
      VEL(I)=VEL(I)+(YFL*2.-1.)*VVEL/100.*VEL(I)
      CALL RANDU(IX,IY,YFL)
      IX=IY
      ANG(I)=ANG(I)+(YFL*2.-1.)*VANG/100.*ANG(I)
C
C        GET THE NUMBER OF COMMANDS IN THE COMMAND STACK
C
      L=NCMDS(I)
      DO 75 J=1,L
C
C        GET THE TYPE OF COMMAND
C
      M=MODCD(J,I)
C
C        CHANGE VALUE OF COMMAND BY CORRECT PERCENTAGE
C
      CALL RANDU(IX,IY,YFL)
      IX=IY
      GO TO (85,86,86,87,87),M
   85 PER=VDIS
      GO TO 88
   86 PER=VVEL
      GO TO 88
   87 PER=VANG
   88 VALC(J,I)=VALC(J,I)+(YFL*2.-1.)*PER/100.*VALC(J,I)
   75 CONTINUE
   77 CONTINUE
C
C        WRITE OUT ALL THE ALTERED DATA
C
      WRITE(6,710)
  710 FORMAT(2X,4HSHIP,5X,15H SHIP NAV. CAP.,5X,7HSTART X,
     X5X,7HSTART Y,5X,11HSTART HEAD.,5X,10HSTART VEL.,5X,
     X10HSTART TIME,5X,8HEND TIME/14X,8H(METERS),9X,8H(METERS),
     X4X,8H(METERS),5X,9H(DEGREES),7X,7H(KNOTS),8X,6H(SECS),
     X9X,6H(SECS)/)
      DO 611 SHIP=1,NSHIPS
      WRITE(6,612)SHIP,NAV(SHIP),PX(SHIP),PY(SHIP),ANG(SHIP),
     X VEL(SHIP),START(SHIP),ENDT(SHIP)
  612 FORMAT(1X,I4,I14,F19.2,F12.2,F13.2,F15.2,F16.2,F14.2)
C
C        CHANGE UNITS
C
      NAV(SHIP)=FLOAT(NAV(SHIP))*AMETER
      PX(SHIP)=PX(SHIP)*AMETER
```

```
C        OR THE SHIP'S REACH HAS BEEN SET TO-1.
C
      OUT(SHIP)=TOTTIM.LT.START(SHIP).OR.TOTTIM.GT.ENDT(SHIP).OR.
     XSHR(SHIP).EQ.-1.
C
C        SKIP CALCULATIONS IF THIS SHIP IS OUT
C
      IF(OUT(SHIP)) GO TO 300
C
C        AT THE BEGINNING OF EACH TIME INTERVAL,SET THE
C        TIME LEFT IN THE INTERVAL TO THE TOTAL INTERVAL
C
      TLINT=TIMINT(K)
C
C        THE SEGMENT IS THE COMMAND OR PHASE BEING EXECUTED
C        TLSEG = TIME LEFT UNTIL END OF SEGMENT
C
    1 IF(TLINT.GT.TLSEG(SHIP))GO TO 100
C
C        IF THERE IS ENOUGH TIME LEFT IN THE SEGMENT MOVE THE
C        THE SHIP FOR THE TIME LEFT IN THE INTERVAL AND DECREASE
C        THE SEGMENT TIME
C
      POSX=PX(SHIP)
      POSY=PY(SHIP)
      CALL MOVE(SHIP,MODE(SHIP),PHASE(SHIP),TLINT,PX(SHIP),PY(SHIP))
      TLSEG(SHIP)=TLSEG(SHIP)-TLINT
      IF(SHR(SHIP).EQ.-2.)GO TO 300
      HDBOX(SHIP)=ATAN2(PY(SHIP)-POSY,PX(SHIP)-POSX)
      GO TO 300
C
C        IF NOT,MOVE THE SHIP FOR THE TIME LEFT IN THE SEGMENT
C
C        GET THE TIME LEFT IN THE INTERVAL AND GET THE NEXT
C        PHASE OR THE NEXT COMMAND
C
  100 CALL MOVE(SHIP,MODE(SHIP),PHASE(SHIP),TLSEG(SHIP),
     X PX(SHIP),PY(SHIP))
      TLINT=TLINT-TLSEG(SHIP)
      IF(PHASE(SHIP).NE.1)GO TO 200
      PHASE(SHIP)=2
C
C        GET THE TIME FOR THIS COMMAND MODE(OR PHASE) AND GO
C        BACK AND TRY TO FINISH THE TIME INTERVAL
C
  150 CALL GETTIM(SHIP,TLSEG(SHIP),MODE(SHIP),PHASE(SHIP))
      GO TO 1
  200 CALL GETCOM(SHIP,OUT)
      IF(OUT(SHIP))SHR(SHIP)=-1.
      IF(OUT(SHIP))GO TO 300
      GO TO 150
  300 CONTINUE
C
C        CHECK IF THERE ARE ANY SHIPS LEFT IN THE HARBOR
C        IF NOT,END THE RUN
```

TABLE 2-2. PROGRAM LISTING FOR MAIN PROGRAM HAR (CONT'D)

```
C
C          INCREMENT THE PRINTER COUNTER AND CHECK FOR PRINTOUT
  390 PRTCNT=PRTCNT+1
      ICHECK=MOD(PRTCNT,PRTFRE(K)).NE.0
C
C          IF NO PRINTOUT THIS TIME, BY-PASS THE PRINT SECTION
C
      IF (ICHECK) GO TO 540
      WRITE(6,111)
C
C          GET THE APPROPRIATE PAGE TITLE
C
C          PRINT THE TITLE
C
      GO TO(701,702,703,704),PCALC
  701 WRITE(6,630)TITLE
      GO TO 705
  702 WRITE(6,630)TITLE,PCOL
      GO TO 705
  703 WRITE(6,630)TITLE,PGRN
      GO TO 705
  704 WRITE(6.630)TITLE,PCOL,PGRN
  705 CONTINUE
  630 FORMAT(1X.22A4,4X,3A4,3X,3A4)
C
C          PRINT THE UNITS
C
      WRITE(6,635)
  635 FORMAT(12X,6H(SECS),5X,8H(METERS),6X,8H(METERS),7X,
     X 7H(KNOTS),5X,9H(DEGREES) //)
      DO 350 SHIP=1,NSHIPS
      IF(OUT(SHIP)) GO TO 350
C
C          CHANGE UNITS FOR PRINTOUT
C
      VCTY=VEL(SHIP)/FEET
      ANGLE=ANG(SHIP)/RAD
      PXM=PX(SHIP)/AMETER
      PYM=PY(SHIP)/AMETER
C
C          IF TURNING, PRINT OUT THE TURNING VELOCITY
C
      IF(MODE(SHIP).GT.3.AND.PHASE(SHIP).EQ.2)VCTY=VT(SHIP)/FEET
C
C          GET THE INDEX FOR THE ARRAY(AMODE) CONTAINING THE MANEUVERS
C
      ITI=MODE(SHIP)
      GO TO(425,425,422,423,424),ITI
  422 ITI=4
      IF(PHASE(SHIP).EQ.2)ITI=3
      GO TO 425
  423 ITI=5
      IF(PHASE(SHIP).EQ.2)ITI=6
      GO TO 425
  424 ITI=5
```

TABLE 2-2. PROGRAM LISTING FOR MAIN PROGRAM HAR (CONT'D)

```
C
C        INSERT THE DISTANCE AND SHIP NUMBERS INTO THE FREQUENCY
C        DISTRIBUTION ROUTINE FOR POSSIBLE C.P.A.
C
      CALL INSERT(NUM,DLTR,DIST,I,J,OCCUR,NINT,RADI)
      IF (ICHECK) GO TO 48
C
C        CHECK IF THIS DISTANCE IS TO BE PRINTED
C
      IF(PRTDIS.EQ.1.AND..NOT.DLTR)GO TO 48
C
C        NM - CONTAINS THE NUMBER OF DISTANCES TO BE PRINTED SO FAR
C
      NM=NM+1
      CHAR(NM)=BLANK
C
C        CONVERT DISTANCE TO METERS
C
      WORK(NM)=DIST/AMETER
C
C        PLACE AN ASTERISK NEXT TO DISTANCE IF DISTANCE IS
C        LESS THAN MAXIMUM C.P.A.
C
      IF(DLTR)CHAR(NM)=STAR
C
C        SAVE THE SHIP NUMBERS FOR PRINTOUT
C
      JWRK(NM)=I
      IWRK(NM)=J
C
C        IF FIVE DISTANCES ARE TO BE PRINTED,PRINT THEM ON ONE LINE
C
      IF(NM.LT.5)GO TO 48
      WRITE(6,530)(JWRK(L),IWRK(L),WORK(L),CHAR(L),L=1,NM)
C
C        START THE NEXT LINE
C
      NM=0
   48 CONTINUE
   47 CONTINUE
  529 IF (ICHECK) GO TO 750
C
C        WRITE ANY DISTANCES THAT ARE LEFT
C
      IF(NM.NE.0)WRITE(6,530)(JWRK(J),IWRK(J),WORK(J),CHAR(J),J=1,NM)
  530 FORMAT((1X,5(2I4,F12.3,A1)))
C
C        IF A TAPE IS DESIRED,WRITE THE NUMBER OF SHIPS,THE TOTAL TIME,
C        THE COORDINATES,THE ANGLE,THE SHIP'S REACH,AND THE OUT VARIABLE
C
      IF(IPLT.NE.1)GO TO 750
      WRITE(9)NSHIPS,TOTTIM
      WRITE(9)(PX(I),PY(I),HDBOX(I),SHR(I),OUT(I),I=1,NSHIPS)
  750 CONTINUE
C
```

3. SUBROUTINES GETTIM AND MOVE

3.1 DESCRIPTION

Subroutines GETTIM and MOVE contain very little logic, and follow the motion equations for each mode.

3.1.1 Subroutine GETTIM

Subroutine GETTIM obtains the time required to complete a command or phase of a command. The equations are straightforward, and are obtained from the motion equations. The equations for the ship's reach, radius of a turn, turning velocity, advance, and time for deceleration are obtained from a previous version of the model.[1] The advance, turning velocity, and radius of the turn are printed in this routine to avoid being bypassed by any print options; i.e., all turn parameters for every turn are printed, no matter what the time increment size or print options may be.

3.1.2 Subroutine MOVE

In subroutine MOVE a call to the dummy subroutine TIDWIN is made. In this model the angle and velocity of the tide and wind are constant for the harbor, and are set in TIDWIN. The calling sequence contains the position (PX, PY), so that the user may, if desired, make the tide and wind vectors a function of position. Control is then sent to the equations for the respective mode. For phase 1 of any mode, the path is a constant path. The variables SAV1 and SAV2 used during turning contain the parameters needed for the integration, namely:

$$\mathrm{SAV1} = \sin\left[\theta_{win} - \theta(t)\right]$$

$$\mathrm{SAV2} = v_T(s)/R.$$

The function GAUSS does the integration and obtains SAV1 and SAV2 from the common block INT. The variable SGN is used to increase or decrease the angle of a ship during a turn for a right or left turn, respectively.

TABLE 3-1. LIST OF VARIABLES FOR SUBROUTINE GETTIM

ADV	Length of the advance before the turn in feet
ADVS	Length of the advance in meters - used for printout
AMETER	Conversion factor for meters to feet
ANG	Array containing the current heading (radians) for each ship
DUMY	Array used to account for locations in common block CHAR not used
FEET	Conversion factor for knots to ft/sec
MODE	Current mode value of ship for which time to execute is required
PI	The constant 3.1415926
PHASE	Current phase of the mode for which the time is required
R	The radius in feet for the turning circle
RSH	The radius in meters of the turning circle
SC	Matrix containing the characteristics of each ship
SHR	Array containing the ship's reach, the distance from the start of a turn to the tangent of the end point of a 180-degree turn. The ship's reach is not used in this model, but is retained for convenience of the user.
TIME	The time in seconds required to execute a command or phase of a command
VAL	The value associated with the mode
VEL	The current (or initial) velocity of a ship in ft/sec
VELS	The current velocity of a ship, equal to VEL(SHIP)

TABLE 3-2. LIST OF VARIABLES FOR SUBROUTINE MOVE

ANGWIN	Wind direction
ANGTID	Tide direction (the set of the current)
COSNEW	Used during turn, the cosine of the new heading
MODE	Mode of the current command
PHASE	Phase of the mode
PX	x-coordinate (NORTH) of the ship before and after the ship movement
PY	y-coordinate (EAST) of the ship before and after the ship movement
SAV1, SAV2	Parameters used in integration during calculation of the wind components
SINNEW	The sine of the new heading during a turn for this time interval
TIME	Time, in seconds, the ship will move in the mode or phase
VSX, VSY	The x- and y-velocity components of the ship
VSXNEW, VSYNEW	The x- and y-velocity components of the ship after the speed change for the time length, TIME
VTID	Velocity of the tide in ft/sec (drift of the current)
VWIN	Velocity of the wind in ft/sec
VWSQ	Velocity of the wind squared

TABLE 3-3. PROGRAM LISTING FOR SUBROUTINE GETTIM (CONT'D)

```
      IF(ADV(SHIP).LE.0.)ADV(SHIP)=0.
C
C        FOR THIS PHASE,VELOCITY IS CONSTANT
C
      TIME=ADV(SHIP)/VELS
      RETURN
C
C        FOR PHASE 2, VAL=NEW COURSE
C
   45 TIME=ABS(VAL(SHIP))*R(SHIP)/VT(SHIP)
      RETURN
      END
```

TABLE 3-4. PROGRAM LISTING FOR SUBROUTINE MOVE (CONT'D)

```
C
C        SET THE NEW VELOCITY COMPONENTS
C
      RETURN
C
C        FOR PHASE 1 OF DECELERATION,GO TO CONSTANT PATH
C
    3 IF(PHASE.EQ.1)GO TO 1
C
C        GET THE NEW VELOCITY AND COMPONENTS
C
      VELNEW=TAN(ATAN(SC(13,SHIP)*VEL(SHIP))-TIME/(SC(11,SHIP)
     X    *SC(13,SHIP)))/SC(13,SHIP)
      SHR(SHIP)=.5*SC(11,SHIP)*ALOG(1.+VELNEW**2)*SC(12,SHIP)+
     X    .5*VELNEW*SC(9,SHIP)
C
C        ADD THE COMPONENTS FOR THE NEW X,Y.
C
      SAV1=SC(13,SHIP)*VEL(SHIP)
      SAV2=SC(13,SHIP)*VELNEW
      SAV3=SC(11,SHIP)/2.*ALOG((1.+SAV1*SAV1)/(1.+SAV2*SAV2))
      WIND=SQRT(SC(8,SHIP)*.0038*VWSQ*ABS(SIN(ANGWIN-ANG(SHIP))))
      PX=PX+COSANG(SHIP)*SAV3+TIME*(VTID*COS(ANGTID)+WIND*COS(ANGWIN))
      PY=PY+SINANG(SHIP)*SAV3+TIME*(VTID*SIN(ANGTID)+WIND*SIN(ANGWIN))
C
C        RESET THE NEW VELOCITY.
C
      VEL(SHIP)=VELNEW
      RETURN
C
C        FOR ADVANCE ,GO TO CONSTANT PATH
C
    4 IF(PHASE.EQ.1)GO TO 1
C
C        SET SWITCH FOR LEFT TURN
C
      SGN=-1.
   10 CONTINUE
C
C        COMPUTE RADS/SEC.
C
      SAV1=VT(SHIP)/R(SHIP)
C
C        GET ANGLE OF WIND OFF THE BOW
C
      SAV2=SIN(ANGWIN-ANG(SHIP))
C
C        GET NEW ANGLE FOR THIS TIME
C
      ANGNEW=ANG(SHIP)+SGN*SAV1*TIME
      IF(ANGNEW.GT.TWOPI)ANGNEW=ANGNEW-TWOPI
      IF(ANGNEW.LT.0.)ANGNEW=ANGNEW+TWOPI
      COSNEW=COS(ANGNEW)
      SINNEW=SIN(ANGNEW)
C
```

4. SUBROUTINE TIDWIN

4.1 DESCRIPTION

This subroutine, as now existing in this document (Table 4-1), is in reality a dummy subroutine. In actual performance of the main program, the user supplies his own subroutine TIDWIN, which contains the correct information whereby to compute the velocity and heading of the tide and wind as a function of the ship's position. This is accomplished by inserting as input the coordinates of the ship's position (x,y), and also by inserting in the main body of the subroutine a table of values of the tide and wind versus a harbor area which will provide the required outputs as functions of the coordinates. This operation is used by subroutine MOVE as required by the ship motion equations.

4.2 PROGRAM LISTING

Table 4-1 provides the listing of subroutine TIDWIN. Note: This table exists in the text in the sense discussed in Section 4.1. For example, values for velocity and heading for tide and wind are arbitrarily assigned the constant value zero. Different values will be provided by the user when he replaces the dummy subroutine with a valid subroutine.

5. SUBROUTINE TRKRD

5.1 DESCRIPTION

a. Subroutine TRKRD converts track data to ship commands. The track data consist of heading, length, and velocity for each track. If possible, the commands generated will move a ship along these tracks. If the track lengths are not long enough to accommodate a command, the length will be overrun by the ship. The command mode MODCD and associated value VALC of each command generated are stored in the common block CMDS. At the end, the number of commands generated (NCMDS) and the command number to be executed [(set equal to 1) (NUMC)] are also stored in the common block.

b. Initially, the number of tracks and the track parameters are read in. The dimension statements limit the number of tracks to 15 per ship. The heading is converted to radians, the velocity to ft/sec, and the length to feet. The initial heading ANG and velocity VEL are set in degrees and knots for printout by the main program. KCM is used for a counter on the commands. The first command, KCM = 1, must be a constant path, so the mode is set to 1 and its value is the length of the track. The velocity for this command is VEL, the current velocity. I is a counter used for checking the track number. When I is equal to the total number of tracks, all the commands have been generated, and control returns to the main program.

c. Figure 5-1 is a logic diagram for converting track data to ship commands. The blocks indicate the sequence of actions taken after decisions are reached with respect to turn maneuvers, acceleration, deceleration, and constant path travel. The following text relates the procedures in these blocks with the steps referred to in the listing for subroutine TRKRD.

d. First, a check is made to see if a turning command will be required. If not, a check for velocity change is made by jumping to statement 20. If a turning command is required, it must be decided whether a right or left turn is in order. If the original heading plus 180 degrees is less than the second heading, a left turn is required. (See Figure 5-2.)

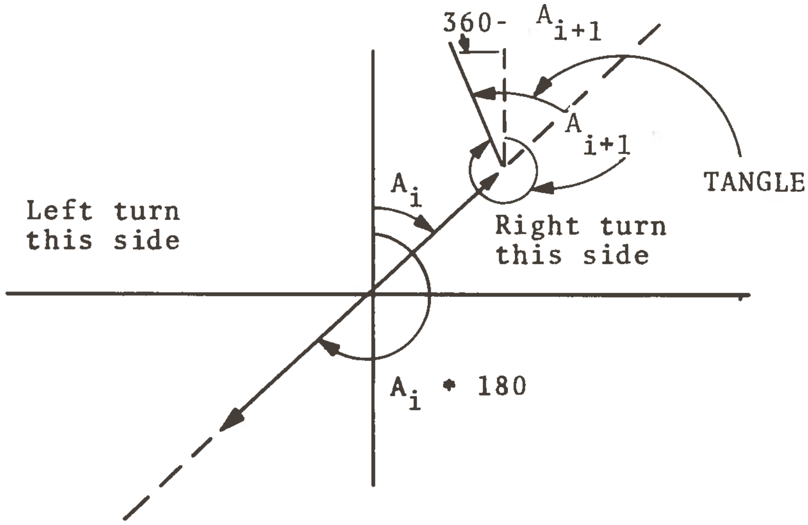

Figure 5-2. Left Turn, Angle Between Successive Tracks (TANGLE) Greater Than Initial Heading A_i

The turning angle is $(360 - A_{i+1})$ plus A_i.

Also, if A_i is greater than A_{i+1}, a left turn is in order, and the turning angle is just $A_i - A_{i+1}$, as shown in Figure 5-3.

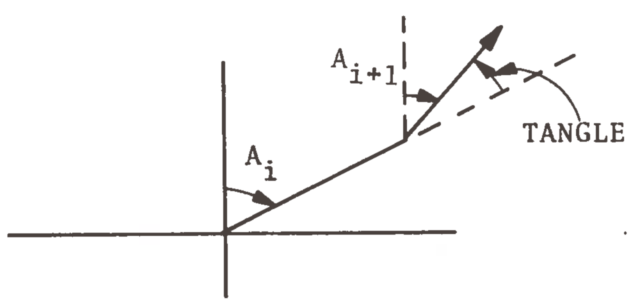

Figure 5-3. Left Turn, Angle Between Successive Tracks (TANGLE) Less Than A_i

TABLE 5-1. LIST OF VARIABLES FOR SUBROUTINE TRKRD

A	Input array containing the heading for each track
ADV	Advance before the turn, the distance required for the ship to respond to change in rudder
D	Input array containing the lengths of each track
DAV	The distance remaining from a previous constant path command after the distance required to turn is subtracted
DIS	The distance required to accelerate or decelerate to a desired speed
KCM	The number of the command last inserted into the command stack. At the end KCM will contain the total number of commands.
MSAV	Variable used to save the mode value of a turn: - MSAV = 4, left turn; MSAV = 5, right turn
REV	Distance required to reverse engines = 1/2 reverse time times v (characteristic SC9)
TANGLE	The turning angle, the angle from the extension of the current track to the next track
TLEFT	The distance from the tangent point of the turning circle with the track to the end of the track
TURN	The turning velocity, less than or equal to track velocity
V	Input array containing the velocities for each track
VSQ	The square of the velocity used in computation of the advance

TABLE 5-2. PROGRAM LISTING FOR SUBROUTINE TRKRD

```
$IBFTC TRK
      SUBROUTINE TRKRD(SHIP)
C
C          ROUTINE TO CONVERT TRACK DATA TO PILOT-TYPE COMMANDS
C
      INTEGER SHIP
      DIMENSION A(15),D(15),V(15)
      COMMON/CHAR/SC(20,15),VEL(15),ANG(15),VAL(15),
     X MODE(15),PHASE(15),SINANG(15),COSANG(15),SHR(15)
      COMMON/CMDS/MODCD(25,15),VALC(25,15),NUMC(15),NCMDS(15)
      DATA RAD,FEET/.0174533,1.6889/,AMETER/3.28083/
      DATA PI/3.14159265/,TWOPI/6.28318531/
C
C          READ IN THE NUMBER OF TRACKS
C
      READ(5,100) NTRKS
  100 FORMAT(I5)
      DO 5 K=1,NTRKS
C
C          FOR EACH TRACK,READ IN THE ANGLE,VELOCITY,AND LENGTH
C
      READ(5,200) A(K),V(K),D(K)
      A(K)=A(K)*RAD
      V(K)=V(K)*FEET
      D(K)=D(K)*AMETER
    5 CONTINUE
      KCM=1
      ANG(SHIP)=A(1)/RAD
      VEL(SHIP)=V(1)/FEET
  200 FORMAT(3F15.5)
      I=1
      MODCD(KCM,SHIP)=1
      VALC(KCM,SHIP)=D(1)/AMETER
C
C          CHECK FOR TURN
C
    6 CONTINUE
      IF(A(I)-A(I+1).EQ.0.)GO TO 20
      IF(A(I)+PI.LT.A(I+1))GO TO 10
      IF(A(I).GT.A(I+1))GO TO 15
      MSAV=5
      TANGLE=A(I+1)-A(I)
      GO TO 11
   10 MSAV=4
      TANGLE=TWOPI-A(I+1)+A(I)
      GO TO 11
   15 MSAV=4
      TANGLE=A(I)-A(I+1)
      GO TO 11
C
C          CHECK FOR CHANGE IN VELOCITY
C
   20 IF(V(I)-V(I+1))40,45,60
   45 IF(D(I+1).LE.0.)GO TO 50
      KCM=KCM+1
```

TABLE 5-2. PROGRAM LISTING FOR SUBROUTINE TRKRD (CONT'D)

```
C
 40 DIS=(V(I+1)*V(I+1)-V(I)*V(I))*SC(14,SHIP)
C
C        ADD THE ACCELERATION COMMAND
C
      KCM=KCM+1
      MODCD(KCM,SHIP)=2
      VALC(KCM,SHIP)=V(I+1)/FEET
      IF((D(I+1)-DIS).LE.0.)GO TO 50
C
C        IF ANY DISTANCE LEFT AFTER ACCELERATION,ADD A CONSTANT
C        PATH FOR THE REST OF THE TRACK
C
      KCM=KCM+1
      MODCD(KCM,SHIP)=1
      VALC(KCM,SHIP)=(D(I+1)-DIS)/AMETER
      GO TO 50
C
C        COMPUTE THE DISTANCE REQUIRED TO DECELERATE
C
 60 DIS=.5*SC(11,SHIP)*ALOG((1.+V(I)*V(I)*SC(12,SHIP))
   X     /(1.+V(I+1)*V(I+1)*SC(12,SHIP)))
C
C        GET THE DISTANCE REQUIRED BY THE REVERSE TIME
C
      REV=.5*SC(9,SHIP)*V(I)
C
C        ADD THE DECELERATION COMMAND
C
      KCM=KCM+1
      MODCD(KCM,SHIP)=3
      VALC(KCM,SHIP)=V(I+1)/FEET
      IF((D(I+1)-DIS-REV).LE.0.)GO TO 50
C
C        IF ANY DISTANCE LEFT,ADD A CONSTANT PATH FOR THE
C        REST OF THE TRACK
C
      KCM=KCM+1
      MODCD(KCM,SHIP)=1
      VALC(KCM,SHIP)=(D(I+1)-DIS-REV)/AMETER
      GO TO 50
800 CONTINUE
C
C        SET THE NUMBER OF COMMANDS,AND THE COMMAND NUMBER
C        TO BE EXECUTED NEXT
C
      NCMDS(SHIP)=KCM
      NUMC(SHIP)=1
600 CONTINUE
      RETURN
      END
```

6. SUBROUTINE GETCOM

6.1 DESCRIPTION

This subroutine obtains the next command in the command stack. At the beginning, a check is made of the number of commands left. If there are no commands left, the ship is to be considered out. The mode and value for the mode are obtained directly from the command stack and stored in the arrays MODE and VAL. The phase of the command is set equal to zero for any mode less than 3 (i.e., for constant path and acceleration) and to 1 for the remaining modes. The command number to be executed next is increased by 1.

6.2 PROGRAM LISTING

Table 6-1 provides the listing of subroutine GETCOM.

7. SUBROUTINE RDCNTR

7.1 DESCRIPTION

Subroutine RDCNTR reads the contour data directly as output by program LINES. Each set of contours (the main contour and its islands) is stored in one column of the matrices XH, YH, S, CT, and ST. The maximum number of points for each contour set is limited by the dimension statements to 150 (i.e., the sum of the number of points in the main contour and the islands cannot be greater than 150). The maximum number of contour sets is 5. The maximum number of islands in any contour set is 5.

7.2 VARIABLES

Table 7-1 provides a list of the main variables in the listing of subroutine RDCNTR.

7.3 PROGRAM LISTING

Table 7-2 provides the listing of subroutine RDCNTR.

TABLE 7-2. PROGRAM LISTING FOR SUBROUTINE RDCNTR

```
$IBFTC CTR
      SUBROUTINE RDCNTR
      COMMON/CNTR/XH(150,5),YH(150,5),S(150,5),CT(150,5),
     X ST(150,5),NCNTR,NPNTS(6,5),NISLDS(5),IFATH(5)
C
C         READ IN THE NUMBER OF CONTOURS
C
      READ(5,1000)NCNTR
      DO 10 I=1,NCNTR
C
C         FOR EACH CONTOUR,READ IN THE DEPTH(FATHOMS) AND THE NUMBER
C         OF ISLANDS
C
      READ(5,1000)IFATH(I),NISLDS(I)
      K=NISLDS(I)+1
C
C         READ IN THE NUMBER OF POINTS FOR THE CONTOUR AND THE ISLANDS
C
      READ(5,1000)(NPNTS(J,I),J=1,K)
C
C         KK IS THE FIRST INDEX FOR THE CONTOUR
C         L IS THE LAST INDEX IN THE SET OF CONTOURS AND ITS ISLANDS
C
      L=0
      DO 20 J=1,K
      KK=L+1
      L=L+NPNTS(J,I)
C
C         READ IN THE ENDPOINTS,SLOPE,SINE,AND COSINE FOR EACH POINT
C
      READ(5,2000)(XH(M,I),YH(M,I),S(M,I),ST(M,I),CT(M,I),M=KK,L)
C
C         RESET THE FIRST INDEX FOR THE NEXT SET
C
   20 CONTINUE
   10 CONTINUE
      RETURN
 1000 FORMAT(10I5)
 2000 FORMAT(5E15.8)
      END
```

TABLE 8-1. LIST OF VARIABLES FOR SUBROUTINE CMDRD

ANG	Initial heading of the ship in degrees
I	Ship number for which data are currently being read in
MODCD	Matrix containing the command modes for each ship
NUMC	Array containing the command number to be executed next for each ship
NCMDS	The total number of commands for each ship
VALC	The value of each command mode stored in MODCD

9. SUBROUTINE PRIPLT

9.1 DESCRIPTION

a. PRIPLT plots, on the printer, a frequency distribution stored in the array OCCUR with NINT values. A maximum distance is required to compute the interval size. The maximum distance is RADI, and the distance range will be from 0 to RADI. The frequency is plotted horizontally and the distance interval vertically.

b. The array LINE is initially set to blanks; then positions 1, 50, and 100 are set to DOT for reference. The value at the first dot is 0, and at the last is the maximum value in the frequency array. The DO 15 loop obtains the maximum value. The interval size (SLOT) is then computed, and this is printed along with the maximum frequency value MAXY.

c. The DO 20 loop prints the NINT lines. The value at the end of the current interval is stored in X, and is incremented each time through the loop. The closest index in the array NINT corresponding to the value of OCCUR is now computed, and is OCCUR/MAXY * 100, which is in proportion to the 100 print positions. The constant, 0.5, ensures correct roundoff when the value is truncated to an integer. The value L is checked to ensure that its range is between 1 and 100. The position in the array LINE is then set to an asterisk, and the interval value, the frequency value, and the line of characters are printed. The line is reset, and the computations are repeated for the next line.

9.2 VARIABLES

Table 9-1 provides a list of the main variables used in the listing of subroutine PRIPLT.

9.3 PROGRAM LISTING

Table 9-2 provides the listing of subroutine PRIPLT.

TABLE 9-2. PROGRAM LISTING FOR SUBROUTINE PRIPLT

```
$IBFTC PRI
      SUBROUTINE PRIPLT(OCCUR,NINT,RADI,BLANK,DOT,STAR)
C
C        ROUTINE TO OUTPUT PRINTER PLOT OF FREQUENCY DISTRIBUTION
C          OCCUR-ARRAY TO BE PLOTTED CONTAINING FREQUENCIES
C          NINT -NUMBER OF INTERVALS FROM ZERO TO RADI
C          BLANK,DOT,STAR - CHARACTERS USED IN PLOT
C
      DIMENSION LINE(100)
      INTEGER OCCUR(1),BLANK,DOT,STAR
      WRITE(6,100)
  100 FORMAT(1H1)
C
C        SET THE FIRST,MIDDLE,AND END POINTS TO A DOT(FOR REFERENCE)
C
      DO 10 I=1,100
   10 LINE(I)=BLANK
      LINE(1)=DOT
      LINE(50)=DOT
      LINE(100)=DOT
C
C        GET THE MAXIMUM FREQUENCY IN ANY CELL
C
      MAXY=-10000.
      DO 15 I=1,NINT
   15 MAXY=MAX0(MAXY,OCCUR(I))
C
C        SET THE SCALING FACTOR(THE LINE IS 100 CHARACTERS LONG)
C
      YMAX=FLOAT(MAXY)/100.
C
C        SET THE INTERVAL SIZE
C
      SLOT=RADI/FLOAT(NINT)
C
C        WRITE THE INTERVAL SIZE AND THE MAXIMUM NUMBER OF
C        OCCURENCES(VALUE AT 100TH CHARACTER)
C
      WRITE(6,150)SLOT,MAXY
  150 FORMAT(60X,29HC.P.A. FREQUENCY DISTRIBUTION /65X,16HINTERVAL SIZE
     X= ,F10.5,9H (METERS)/65X,31HMAXIMUM NUMBER OF OCCURENCES = ,I4/)
      X=0
      DO 20 I=1,NINT
C
C        GET THE VALUE OF THE CURRENT INTERVAL
C
      X=X+SLOT
C
C        SCALE THE VALUE TO CLOSEST CHARACTER NUMBER
C
      L=FLOAT(OCCUR(I))/YMAX+.5
C
C        TAKE CARE OF END POINTS
C
      IF(L.EQ.0)L=1
```

10. SUBROUTINES INSERT AND ENDINS

10.1 DESCRIPTION

10.1.1 Subroutine INSERT

Subroutine INSERT is used to compute the closest point of approach (CPA) for pairs of ships, and to accumulate these for a frequency distribution. The CPA is the minimum distance of approach of two ships; therefore, distances must be saved at each time step until the minimum is reached. For a large number of ships, the size of the arrays required would be prohibitive, and thus, restrictions were applied:

a. Ships a distance DMAX apart or greater will not be checked.

b. A maximum of 100 pairs of ships for any time step will be checked.

For a maximum of 100 pairs of ships, it is required to be able to distinguish which pairs are being checked. The number UNQ is used for this purpose. If ship I is being checked against ship J, the number is set at I*1000+J. These numbers are stored in the array IWH. On entry to INSERT, the unique number for ships I and J is computed, and this number is checked against those stored in IWH. If the number is there, a minimum is already being sought for this pair. If the current distance DIST is less than the previous distance saved, the current distance is the new minimum. If the current distance DIST is greater than the distance saved, then the distance saved is the minimum required, and one is added to the correct interval in the array OCCUR. The total number being checked is decreased by one, and the arrays containing the unique numbers and their corresponding distances are pushed down to delete further reference to this pair. Statement 50 is used to ensure that the distance between the ships being checked is greater than DMAX before adding to the array OCCUR; else, the pair could possibly be again checked during the next time step. If the pair

TABLE 10-1. LIST OF VARIABLES FOR SUBROUTINES INSERT AND ENDINS

DIS	Array containing the minimum distance thus far obtained for a pair of ships
DIST	The current distance being checked for minimum value against the DIS array. DIST will replace DIS if it is less.
DLTR	Logical variable set true by main program when the distance between two ships is less than RADI (or the equivalent, DMAX, used in this routine)
DMAX	Maximum distance apart two ships will be checked for closest point of approach
I,J	Current ship numbers being checked
IWH	Array containing numbers that signify which pair of ships is currently being checked for closest point of approach
LL	Index for array location in array IWH that contains the number UNQ for current pair of ships if one exists
NINT	Number of intervals to be used in the range 0 to DMAX for the frequency plot
NUM	Total number of pairs of ships currently being checked for closest point of approach
OCCUR	Array containing the total number of CPAs accumulated for each interval between 0 and DMAX. A maximum of 100 intervals is allowed.
UNQ	A number used to distinguish pairs of ships. The number is computed as I*1000+J, where I and J are the ship numbers.

TABLE 10-2. PROGRAM LISTING FOR SUBROUTINE INSERT (CONT'D)

```
      IF(DIST.GT.DIS(LL))RETURN
C
C        IF NOT, THIS IS NEW MINIMUM
C
      DIS(LL)=DIST
      RETURN
C
C        IF DISTANCE IS GREATER THAN MAX.,THE PREVIOUSLY SAVED POINT IS
C        THE MINIMUM. SCALE THE DISTANCE AND ADD ONE TO CORRECT SLOT
C
  200 L=DIS(LL)*FLOAT(NINT)/DMAX+1.
      OCCUR(L)=OCCUR(L)+1
      IF(LL.EQ.NUM)GO TO 80
C
C        REDUCE THE NUMBER OF PAIRS BEING CHECKED AND CLOSE UP
C        THE ARRAYS
C
      NM=NUM-1
      DO 70 L=LL,NM
      IWH(L)=IWH(L+1)
   70 DIS(L)=DIS(L+1)
   80 NUM=NUM-1
      RETURN
      END
```

11. FUNCTION GAUSS

11.1 DESCRIPTION

The Gaussian integration formula and the required constants, H and WTS, are given in the Handbook of Mathematical Functions (Abramowitz and Stegun),[2] Equation 25.4.30 and Table 25.4. The following formula is an approximation of Equation 25.4.30.

$$\int_a^b f(x)dx = \frac{b-a}{2} \sum_{i=1}^{n} w_i f(y_i)$$

where $y_i = \frac{b-a}{2} x_i + \frac{b+a}{2}$

and the x_i and w_i are given in the table. The weights, w_i, and the abscissas, x_i, are symmetric about the midpoint, and therefore only n/2 are required. The midpoint of the interval is given by D in the program, and is equal to (a+b)/2. The sum may now be written as:

$$\sum_{i=1}^{n/2} w_i (f(u_i) + f(v_i))$$

where $u_i = \frac{b+a}{2} + \frac{b-a}{2} x_i$

$$v_i = \frac{b+a}{2} - \frac{b-a}{2} x_i$$

This sum is given by SUM, and the value of the integral is (B-A/2) *SUM. The function to be integrated is

F(T) = SQRT(ABS(SIN(SAVE1 - SAVE2*T))).

TABLE 11-2. PROGRAM LISTING FOR FUNCTION GAUSS

```
$IBFTC GAUS
      FUNCTION GAUSS(A,B)
C
C        ROUTINE TO INTEGRATE THE FUNCTION USED IN THE TIDE
C        AND WIND CALCULATIONS(METHOD OF GAUSSIAN QUADRATURE)
C
      COMMON/INT/SAVE1,SAVE2
      DIMENSION H(10),WTS(10)
C
C        STORE THE ZEROS OF THE LEGENDRE POLYNOMIALS
C
      DATA H/    .0765265,.2277859,.3737061,.5108670,.6360537,
     X           .7463319,.8391170,.9122344,.9639720,.9931286/
C
C        STORE THE WEIGHTS
C
      DATA WTS/ .1527534,.1491730,.1420961,.1316886,.1181945,
     X          .1019301,.0832767,.0626720,.0406014,.0176140/
      SUM=0.
C        CHANGE LIMITS TO 0. AND 1.
C
C
      C=.5*(B-A)
      D=.5*(B+A)
      DO 1 J=1,10
      T=C*H(J)
C
C        ROOTS OF LEGENDRE POLYNOMIALS ARE SYMMETRIC  ABOUT D
C
      X=D+T
      XM=D-T
C
C        EVALUATE THE FUNCTION AT THESE POINTS
C
      F1=SIN(SAVE1-SAVE2*X)
      F1=ABS(F1)
      F2=SIN(SAVE1-SAVE2*XM)
      F2=ABS(F2)
C
C        SUM THE PRODUCTS WTS(J)*F(I)
C
    1 SUM=SUM+WTS(J)*(SQRT(F1)+SQRT(F2))
C
C        SCALE THE RESULTS
C
      GAUSS=SUM*C
      RETURN
      END
```

TABLE 12-1. PROGRAM LISTING FOR SUBROUTINE RANDU

FOR 32-BIT WORD

```
      SUBROUTINE RANDU(IX,IY,YFL)                                       HAR19010
      IY=IX*65539                                                       HAR19020
      IF(IY)5,6,6                                                       HAR19030
    5 IY=IY+2147483647+1                                                HAR19040
    6 YFL=IY                                                            HAR19050
      YFL=YFL*.4656613E-9                                               HAR19060
      RETURN                                                            HAR19070
      END                                                               HAR19080
```

FOR 36-BIT WORD

```
$IBFTC RANO                                                             HAR19010
      SUBROUTINE RANDU(IX,IY,YFL)                                       HAR19020
      IY=IX*65539                                                       HAR19030
      IF(IY)5,6,6                                                       HAR19040
    5 IY=IY+34359738367+1                                               HAR19050
    6 YFL=IY                                                            HAR19060
      YFL=YFL*.2910383E-10                                              HAR19070
      RETURN                                                            HAR19080
      END                                                               HAR19090
```

FOR 60-BIT WORD

```
      SUBROUTINE RANDU(IX,IY,YFL)                                       HAR19010
      IY=IX*65539                                                       HAR19020
      IF(IY)5,6,6                                                       HAR19030
    5 IY=IY+576460752303423487+1                                        HAR19040
    6 YFL=IY                                                            HAR19050
      YFL=YFL*.17347234759768E-17                                       HAR19060
      YFL=YFL*2000.                                                     HAR19070
      RETURN                                                            HAR19080
      END                                                               HAR19090
```

(1) ALPRG

(2) PGNDN

(3) ELIPS

(4) EPSHL

(5) ANGLES

(6) ERINT

(7) R

(8) EXPX

Subroutine ALPRG just accumulates the logs of probability of no-grounding at each time step.

b. The probability of grounding is computed as the volume above land under the probability surface. Land is considered as anything outside the safe depth contour and inside the islands. The depth contour is decreased in area by the amount that a ship would travel during the current time step, which is the ship's length/2 + velocity * the time interval. The islands are not altered in these routines, although they should be increased, for completeness, by the same amount as the contours were decreased, by the same reasoning.

c. Refer to Figure 13-1, and consider a ship to be traveling outward from the safe area bounded by endpoints (XSH_i, YSH_i) (not yet determined). As the ship approaches a given outer contour bounded by endpoints (XH_i, YH_i), and the probability of grounding increases (since the outer contour is considered to be in the vicinity of land), the new contour boundaries for safe travel are determined by decreasing the boundaries of the outer contour (XH_i, YH_i). The amount of reduction equals the distance that the ship would travel in the next time step plus half the ship's length.

d. Now consider the ship within the safe area approaching an island in the same area, also as shown in Figure 13-1. In this case, determination of safe travel is the reverse procedure from that described in c. The island is now considered to be increased by the distance that the ship would travel in the next time step

TABLE 13-1. LIST OF VARIABLES FOR PROBABILITY SUBROUTINES

ADD	Value of probability density curve given the angle, correlation coefficient, standard deviations, and a distance. The value is from the given distance to infinity.
ANGMAX	The angle between the ship's position and the endpoints of a contour line for which the ship is considered on the line.
DELTA	The increment in angle used for integration. For increasing angle between end points of a contour line, DELTA is positive. For decreasing angle DELTA is negative.
DIFF	The angle between the endpoints of a contour line and the ship's position. DIFF is negative for decreasing angle in a clockwise direction.
DTHETA	The absolute size of the increment in angle = 1./factor radians
FACTOR	Constant set to 100 in program. It determines the size of DTHETA. For FACTOR = 100, DTHETA is 0.57 degree.
I	The current ship number
ICT	The contour number being used as a safe harbor
IFATH	Array containing the depth level of each contour in fathoms
INCRE	The number of angle increments (DTHETAs needed to complete a contour line)
M	The contour or island being processed for the ICT contour set. M ranges from 1 (main contour) to the number of islands + 1

TABLE 13-1. LIST OF VARIABLES FOR PROBABILITY SUBROUTINES (CONT'D)

VEL	The velocity of the ship
XH,YH	The endpoints of the contour lines and islands
XSH, YSH	The endpoints of the contour lines obtained by decreasing the original contour by the amount the ship will travel during the current time step.

navigational error, which is the sum of the on-board error and the error due to harbor conditions and equipment. The harbor error is defined by a harbor index, NI. Each unit of NI signifies a 30-foot navigational error. NI is set to 5 in this model. More realistically, NI should be a function of position. For example, in an area where buoys are located, the harbor error would be nearly zero. On the other hand, during adverse weather conditions such as foggy weather, the harbor error would be greater. Referring to the subroutine ELIPS (Table 13-3) and to Figure 13-2, note that SIGX (σ_x) is set equal to SIGY(σ_y). This is to say that the error in one direction is the same as the error in another direction, or harbor error is equal in all directions. Since σ_x equals σ_y, and all sigmas are referenced to the mean position (UX, UY) of the ship, then the 3σ ring is a circle. The correlation coefficient in this case (R) was set to zero, i.e., errors are independent of direction.

b. The DO 1000 loop obtains the contour to be used for this ship. If the contours are exhausted before the correct depth is reached, the last contour (greatest depth for contours are read-in in order of depth) is used. ICT contains the contour number.

c. The assignment statement will allow the variables NT and SGN to be changed when integrating around islands. For obtaining the area around the main contour SGN = -1, and for the islands SGN = 1. This allows the accumulated area, SUM, to be positive for the main contour by adding the segments and to decrease by subtracting the segments around an island. SGN is -1 for the main contour, because in going around clockwise DELTA, the angle increment, is negative.

d. The main contour is then decreased by subroutine EPSHL (Table 13-9) and the new points stored in XSH, YSH. The angles to these points are computed in subroutine ANGLES (Table 13-10). The difference in angles, going clockwise, is then computed. If the difference is positive, the area outside this line is subtracted from the accumulated sum, segment by segment. If the difference is negative, the area is added. The reverse is true of the islands; thus, the use of the variable SGN. The net result is the volume

TABLE 13-2. PROGRAM LISTING FOR SUBROUTINE ALPROB

```
$IBFTC ALP
      SUBROUTINE ALPROB(OUT,PX,PY,SC,VEL,NS,ALP,NAV)
      DIMENSION NAV(1)
      LOGICAL OUT(1)
      DIMENSION PX(1),PY(1),SC(20,1),VEL(1),ALP(1)
C
C        CHECK FOR MORE THAN  ONE  SHIP
C
      IF(NS.EQ.1) RETURN
      II=NS-1
      DO 100 I=1,II
C
C        BY-PASS CALCULATIONS IF SHIP IS OUT
C
      IF(OUT(I)) GO TO 100
C
C        GET SIGX,SIGY FOR THE FIRST SHIP
C
      CALL ELIPS(I,PX(I),PY(I),SIGXI,SIGYI,RI,NAV)
      KK=I+1
C
C        CHECK AGAINST ALL OTHER SHIPS
C
      DO 200 J=KK,NS
      IF(OUT(J))GO TO 200
C
C        IF PROB(COLLISION) HAS BEEN  ONE ,BY-PASS CALCULATIONS
C
      IF((ALP(I).LT.-100.0).AND.(ALP(J).LT.-100.0)) GOTO 200
C
C        GET SIGX,SIGY FOR THE SECOND SHIP
C
      CALL ELIPS(J,PX(J),PY(J),SIGXJ,SIGYJ,RJ,NAV)
C
C        GET PROBABILITY OF NO COLLISION
C
      Y=1.0-PCOLIJ(PX(I),PX(J),PY(I),PY(J),SIGXI,SIGXJ,
     1SIGYI,SIGYJ,RI,RJ,SC(1,I),SC(1,J),VEL(I),VEL(J))
C
C        IF PROB(COLLISION) IS ZERO,DO NOT ACCUMULATE
C
      IF(Y.GE.1.0) GOTO 200
C
C        IF PROB. IS BETWEEN ZERO AND ONE, ACCUMULATE
C
      IF(Y.GT.0.0) GOTO 10
      ALP(I)=-101.0
C
C        IF PROB(COLLISION) IS 1.,SET LOG = MINUS INFINITY
C
      ALP(J)=-101.0
      GOTO 200
   10 Y=ALOG(Y)
      ALP(I)=ALP(I)+Y
      ALP(J)=ALP(J)+Y


  200 CONTINUE
  100 CONTINUE
      RETURN
      END
```

TABLE 13-3. PROGRAM LISTING FOR SUBROUTINE ELIPS

```
$IBFTC ELIP
      SUBROUTINE ELIPS(I,PX,PY,SIGX,SIGY,R,NAV)
C
C        SUBROUTINE TO COMPUTE STANDARD DEVIATIONS,SIGX,SIGY FOR SHIP I
C
      DIMENSION NAV(15)
C
C        CORRELATION COEFFICIENT IS SET TO ZERO
C
      R=0.0
C
C        NAV INDEX FOR HARBOR IS SET TO 5
C
      NI=5
C
C        SIGX IS  1/3 RADIUS OF 3-SIGMA RING
C
      SIGX=FLOAT(NAV(I)+30*NI)/3.
      SIGY=SIGX
      RETURN
      END
```

Note: Refer to Section 13.1.3 for discussion of ELIPS and its relationship to main subroutine PGNDN.

TABLE 13-4. PROGRAM LISTING FOR FUNCTION PCOLIJ

```
$IBFTC PCOL                                                             HAR14630
      FUNCTION PCOLIJ(UXI,UXJ,UYI,UYJ,SIGXI,SIGXJ,SIGYI,                HAR14640
     1SIGYJ,RI,RJ,SLI,SLJ,VI,VJ)                                        HAR14650
C                                                                       HAR14660
C        SUBROUTINE TO COMPUTE THE PROB(COLLISION) BETWEEN TWO SHIPS    HAR14670
C                                                                       HAR14680
C                                                                       HAR14690
C        UXI,UYI - MEAN POSITION OF FIRST SHIP                          HAR14700
C        UXJ,UYJ - MEAN POSITION OF SECOND SHIP                         HAR14710
C        SIGXI,SIGYI,SIGXJ,SIGYJ - STANDARD DEVIATION IN POSITION FOR   HAR14720
C                                  EACH DENSITY CURVE                   HAR14730
C        RI,RJ - CORRELATION COEFFICIENT FOR EACH CURVE                 HAR14740
C        VI,VJ - VELOCITY OF EACH SHIP                                  HAR14750
C        SLI,SLJ - LENGTH OF EACH SHIP                                  HAR14760
C                                                                       HAR14770
      COMMON/INTVL/TAU                                                  HAR14780
      UDX=UXI-UXJ                                                       HAR14790
      UDY=UYI-UYJ                                                       HAR14800
      SIGDX=SIGXI*SIGXI+SIGXJ*SIGXJ                                     HAR14810
      SIGDY=SIGYI*SIGYI+SIGYJ*SIGYJ                                     HAR14820
      SIGDX=SQRT(SIGDX)                                                 HAR14830
      SIGDY=SQRT(SIGDY)                                                 HAR14840
      RD=(RI*SIGXI*SIGYI+RJ*SIGXJ*SIGYJ)/(SIGDX*SIGDY)                  HAR14850
C                                                                       HAR14860
C        GET ALLOWABLE DISTANCE BETWEEN SHIPS FOR NO-COLLISION          HAR14870
C                                                                       HAR14880
      D=(SLI+SLJ)/2.+(VI+VJ)*TAU                                        HAR14890
C                                                                       HAR14900
C        GET PROBABILITY THAT DISTANCE BETWEEN SHIPS IS LESS THAN D     HAR14910
C                                                                       HAR14920
      PCOLIJ=PDLTD(RD,UDX,UDY,SIGDX,SIGDY,D)                            HAR14930
      RETURN                                                            HAR14940
      END                                                               HAR14950
```

TABLE 13-6. PROGRAM LISTING FOR FUNCTION NDTR

```
$IBFTC NDT                                                      HAR15630
      FUNCTION NDTR(X)                                          HAR15640
      REAL NDTR                                                 HAR15650
      AX=ABS(X)                                                 HAR15660
      IF(AX.GT.6.0) GOTO 3                                      HAR15670
      T=1.0/(1.0+0.2316419*AX)                                  HAR15680
      D=0.3989423*EXP(-X*X/2.0)                                 HAR15690
      NDTR=1.0-D*T*((((1.330274*T-1.821256)*T+                  HAR15700
     11.781478)*T-0.3565638)*T+0.3193815)                       HAR15710
    4 IF(X) 1,2,2                                               HAR15720
    1 NDTR=1.0-NDTR                                             HAR15730
    2 RETURN                                                    HAR15740
    3 NDTR=1.0                                                  HAR15750
      GOTO 4                                                    HAR15760
      END                                                       HAR15770
```

TABLE 13-8. PROGRAM LISTING FOR FUNCTION PGNDN

```
$IBFTC PGND
      FUNCTION PGNDN(PX,PY,I,NAV,SC1,SC3,VEL)
C
C        FUNCTION COMPUTING THE PROBABILITY OF GROUNDING
C
C        THE POINTS OF THE CONTOURS(XH,YH) MUST BE STORED CLOCKWISE
C           (PX,PY) IS THE MEAN POSITION OF SHIP I
C            SC1  IS THE LENGTH OF THE SHIP
C            SC3  IS THE DRAFT OF THE SHIP
C
      DIMENSION NAV(1)
      COMMON/CNTR/XH(150,5),YH(150,5),S(150,5),CT(150,5),
     X ST(150,5),NCNTR,NPNTS(6,5),NISLDS(5),IFATH(5)
      DIMENSION TT(150),XSH(150),YSH(150)
C
C        STORE CONSTANTS
C        ANGMAX IS MAXIMUM ANGLE BETWEEN POSITION OF SHIP AND
C        END POINTS OF THE CONTOUR LINE. FOR AN ANGLE GREATER THAN
C        THIS,THE POSITION IS CONSIDERED ON THE LINE
C
C        FACTOR SETS THE SIZE OF THE INCREMENT IN ANGLE = 1./FACTOR
C
      DATA TPI/6.283185/,PI/3.1415926/,ANGMAX/2.75/,FACTOR/100./
      UX=PX
      UY=PY
      CALL ELIPS(I,UX,UY,SIGX,SIGY,RHO,NAV)
      ICT=0
      IF(NCNTR.EQ.1)GO TO 900
      MCNTR=NCNTR-1
      DO 1000 J=1,MCNTR
      ICT=J
      IF(SC3.GE.FLOAT(IFATH(J))*6. .AND.
     X   SC3.LT.FLOAT(IFATH(J+1))*6.)GO TO 15
 1000 CONTINUE
  900 ICT=ICT+1
   15 CONTINUE
C
C        M IS THE CONTOUR(OR ISLAND) BEING PROCESSED
C        NT IS THE NUMBER OF TIMES THROUGH THE INTEGRATION LOOP(NT IS 1
C        FOR THE MAIN CONTOUR AND IS EQUAL TO THE NUMBER OF ISLANDS
C        WHEN INTEGRATING OVER THE ISLANDS
C        SUM IS THE VALUE OF THE PROBABILITY CURVE OVER THE WHOLE AREA
C
      M=0
      NT=1
      SUM=0.
      ASSIGN 55 TO IN
      SGN=-1.
      N=1
   18 DO 30 J=1,NT
      M=M+1
C
C        GET THE NUMBER OF LINES BOUNDING THE AREA
C
      NL=NPNTS(M,ICT)-1
```

TABLE 13-8. PROGRAM LISTING FOR FUNCTION PGNDN (CONT'D)

```
C
      THETA=TT(L)-DELTA
      LL=N+L-1
      DO 10 K=1,INCRE
C
C        SET THE CURRENT ANGLE FROM THE START POINT OF THE LINE
C
      THETA=THETA+DELTA
      THETA=AMOD(THETA,TPI)
      IF(THETA.LT.0.)THETA=THETA+TPI
C
C        GET THE LENGTH OF THE LINE FROM THE POSITION OF THE SHIP
C        TO THE CURRENT POINT ON THE LINE
C
      RR=R(UX,UY,YSH(L),XSH(L),S(LL,ICT),THETA)
C
C        GET THE VALUE OF THE PROBABILITY INTEGRAL FOR THIS SEGMENT
C
      ADD=ERINT(THETA,RHO,SIGX,SIGY,RR)
C
C        ADD OR SUBTRACT THE AREA(ADD IF CLOCKWISE,SUBTRACT IF NOT)
C
C
C        COMPLETE THE INTEGRAL BY MULTIPLYING BY DTHETA
C
      SUM=SUM+ADD*SGN*DELTA
   10 CONTINUE
      GO TO 20
   22 SUM=SUM+.5
   20 CONTINUE
C
C        SET THE START INDEX FOR THE NEXT AREA
C
      N=N+NL+1
   30 CONTINUE
C
C        DO THE ISLANDS IF NOT YET DONE
C
      GO TO IN,(55,60)
   55 IF(NISLDS(ICT).LT.1)GO TO 60
      ASSIGN 60 TO IN
C
C        SET THE PARAMETERS FOR THE NEXT INTEGRATION
C
      NT=NISLDS(ICT)
      SGN=1.
      GO TO 18
C
C        PROBABILITY OF GROUNDING IS FINAL SUM
C
   60 CONTINUE
      PGNDN=ABS(SUM)
      RETURN
      END
```

TABLE 13-10. PROGRAM LISTING FOR SUBROUTINE ANGLES

```
$IBFTC ANGL                                                             HAR12330
      SUBROUTINE ANGLES(NL,THETA,XSH,YSH,UX,UY)                         HAR12340
C                                                                       HAR12350
C        SUBROUTINE TO COMPUTE THE ANGLES(THETA) FROM THE MEAN POSITION HAR12360
C        OF THE SHIP(UX,UY) TO THE END POINTS OF THE SAFE HARBOR        HAR12370
C        LINES(XSH,YSH)                                                 HAR12380
C        NL IS THE NUMBER OF LINES IN THE CONTOUR                       HAR12390
C                                                                       HAR12400
      DIMENSION THETA(1),XSH(1),YSH(1)                                  HAR12410
      DATA TPI/6.283185307/                                             HAR12420
      DO 10 L=1,NL                                                      HAR12440
C                                                                       HAR12430
C        GET THE DIFFERENCE IN X AND Y                                  HAR12450
C                                                                       HAR12460
      DX=XSH(L)-UX                                                      HAR12470
      DY=YSH(L)-UY                                                      HAR12480
C                                                                       HAR12490
C        COMPUTE THE ANGLE FROM THE ARC-TANGENT                         HAR12500
C                                                                       HAR12510
      THETA(L)=ATAN2(DY,DX)                                             HAR12520
C                                                                       HAR12530
C        CHECK IF ANGLE IS GREATER THAN TWO*PI OR LESS THAN ZERO        HAR12540
C                                                                       HAR12550
      THETA(L)=AMOD(THETA(L),TPI)                                       HAR12560
      IF(THETA(L).LT.0.0) THETA(L)=THETA(L)+TPI                         HAR12570
   10 CONTINUE                                                          HAR12580
C                                                                       HAR12590
C        SET THE LAST ANGLE EQUAL TO THE FIRST                          HAR12600
C                                                                       HAR12610
      THETA(NL+1)=THETA(1)                                              HAR12620
      RETURN                                                            HAR12630
      END                                                               HAR12640
```

TABLE 13-12. PROGRAM LISTING FOR FUNCTION R

```
$IBFTC RF
      FUNCTION R(UX,UY,YS,XS,S,T)
C
C        ROUTINE TO COMPUTE THE LENGTH OF THE LINE FROM THE
C        MEAN POSITION OF THE SHIP(UX,UY) TO THE LINE WITH
C        START POINT(XS,YS) AND ANGLE T
C
      IF(S.GT.1.E13)GO TO 40
      R=((YS-UY)-S*(XS-UX))/(SIN(T)-S*COS(T))
      RETURN
   40 R=(XS-UX)/COS(T)
      RETURN
      END
```

14. SUBROUTINE MISS

14.1 DESCRIPTION

a. Subroutine MISS was written as a demonstration of a simple decision routine. It is not in the program, but it was used as an example of collision avoidance by two ships. (When used, the subroutine is inserted in the program at the beginning of a time-step. For example, the subroutine can be inserted at the beginning of the DO 300 loop in the main program HAR (Table 2-2) after card No. HAR03970.) The decision rule is to have each ship, if a collision situation is detected, to turn right 10 degrees, then left 10 degrees to avoid collision, and then to follow the commands in the command stack. A collision situation is needed to cause this series of maneuvers to be initiated. The criteria used here included the following:

1. If the two ships will reach the intersection of their tracks in the same minute, action is required.
2. Both ships must be at least five minutes from the intersection point.

These criteria are provided for in the DO 20 loop of the program listing for subroutine MISS. DISI and DISJ are the distances to the intersection point for ships I and J. BI and BJ are the intercepts for the two tracks. XINT, YINT are the coordinates for the intersection point.

b. If the collision criteria are satisfied, three commands are to be inserted and executed:

1. Turn right 10°
2. Turn left 10°
3. Straight ahead for the rest of the last track, and then execute commands as before. The commands above the one being executed are saved in the arrays SAVE and ISAVE. The three new commands are inserted after the current command.

TABLE 14-1 PROGRAM LISTING FOR SUBROUTINE MISS

```
$IBFTC MI
      SUBROUTINE MISS(OUT,PX,PY,HDBOX,NSHIPS,MODE,VEL,TLSEG)
C
C        DEMONSTRATION SUBROUTINE TO SHOW A METHOD OF
C        CHANGING THE COMMANDS IN THE COMMAND STACK
C
      COMMON/CMDS/MODCD(25,15),VALC(25,15),NUMC(15),NCMDS(15)
      DIMENSION SAVE(25),ISAVE(25)
      LOGICAL OUT(1)
      DIMENSION PX(1),PY(1),HDBOX(1),MODE(1),VEL(1),TLSEG(1)
      DATA RAD/.0174533/
C
C        IF ONLY ONE SHIP ,RETURN
C
      IF(NSHIPS.EQ.1) RETURN
      II=NSHIPS-1
      DO 10 I=1,II
C
C        IF NOT CONSTANT PATH,BY-PASS
C
      IF(MODE(I).NE.1) GO TO 10
C
C        IF OUT, BY-PASS
C
      IF(OUT(I)) GO TO 10
C
C        CHECK AGAINST ALL SHIPS
C
      KK=I+1
      DO 20 J=KK,NSHIPS
C
C        GET THE SLOPES OF THE PATHS
C
      SLI=TAN(HDBOX(I))
      SLJ=TAN(HDBOX(J))
C
C        GET EQUATIONS OF PATH
C
      BI=PY(I)-SLI*PX(I)
      BJ=PY(J)-SLJ*PX(J)
C
C        GET INTERSECTION POINT
C
      XINT=(BJ-BI)/(SLI-SLJ)
      YINT=SLI*XINT+BI
C
C        COMPUTE DISTANCE TO INTERSECTION POINT FOR SHIP I
C
      DY=PY(I)-YINT
      DX=PX(I)-XINT
      DISI=SQRT(DY*DY+DX*DX)
C
C        COMPUTE DISTANCE TO INTERSECTION POINT FOR SHIP J
C
      DY=PY(J)-YINT
```

TABLE 14-1 PROGRAM LISTING FOR SUBROUTINE MISS (CONT'D)

```
      LL=NCMDS(J)                                                       MISS0555
      IS=0                                                              MISS0560
      DO 300 K=JJ,LL                                                    MISS0565
      IS=IS+1                                                           MISS0570
      SAVE(IS)=VALC(JJ,J)                                               MISS0575
  300 ISAVE(IS)=MODCD(JJ,J)                                             MISS0580
      NCMDS(J)=NCMDS(J)+3                                               MISS0585
      MODCD(JJ,J)=5                                                     MISS0590
      VALC(JJ,J)=10.*RAD                                                MISS0595
      MODCD(JJ+1,J)=4                                                   MISS0600
      VALC(JJ+1,J)=10.*RAD                                              MISS0605
      MODCD(JJ+2,J)=1                                                   MISS0610
      VALC(JJ+2,J)=VEL(J)*TLSEG(J)                                      MISS0615
      IST=JJ+2                                                          MISS0620
      DO 320 K=1,IS                                                     MISS0625
      IST=IST+1                                                         MISS0630
      VALC(IST,J)=SAVE(K)                                               MISS0635
  320 MODCD(IST,J)=ISAVE(K)                                             MISS0640
      TLSEG(J)=0.                                                       MISS0645
   20 CONTINUE                                                          MISS0650
   10 CONTINUE                                                          MISS0655
      RETURN                                                            MISS0660
      END                                                               MISS0665
```

15.2 VARIABLES

Table 15-1 provides a list of the main variables used in the program to generate the sines, cosines, and slopes of the depth contours.

15.3 PROGRAM LISTING

Table 15-2 provides a listing of the program referred to in Section 15.2.

TABLE 15-2. PROGRAM LISTING FOR PROGRAM TO GENERATE SINES, COSINES, AND SLOPES OF DEPTH CONTOURS

```
C        ROUTINE FOR COMPUTING THE SINE,COSINE,AND SLOPES
C        OF THE DEPTH CONTOURS
C
      DIMENSION XH(150),YH(150),S(150),ST(150),CT(150) ,NPOINT(6)
C
C        READ IN THE NUMBER OF CONTOURS,COORDINATES OF THE ORIGIN,
C        SCALING FACTOR AND LOGICAL UNIT NUMBER FOR THE PUNCH(IF
C        NO PUNCH AVAILABLE SET IUNIT=0)
C
      READ(5,100)NCNTRS,XCOORD,YCOORD,SCALE,IUNIT
C
C        FOR EACH CONTOUR ,READ IN THE NUMBER OF POINTS(CLOSED LOOP)
C        AND THE COORDINATES(READ IN CLOCKWISE AROUND CONTOUR)
C
      IF(IUNIT.NE.0)WRITE(IUNIT,10)NCNTRS
      DO 20 K=1,NCNTRS
      READ(5,10)IF,NIS
      IF(IUNIT.NE.0)WRITE(IUNIT,10)IF,NIS
      II=NIS+1
      READ(5,10)(NPOINT(J),J=1,II)
      IF(IUNIT.NE.0)WRITE(IUNIT,10)(NPOINT(J),J=1,II)
      DO 30 J=1,II
      NPNTS=NPOINT(J)
      READ(5,300)(XH(I),YH(I),I=1,NPNTS)
C
C        SCALE THE POINTS
C
      DO 650 L=1,NPNTS
      XH(L)=(XH(L)-XCOORD)*SCALE
      YH(L)=(YH(L)-YCOORD)*SCALE
  650 CONTINUE
      L=NPNTS-1
      DO 60 I=1,L
C
C        GET THE DIFFERENCE IN X AND Y
C
      DX=XH(I+1)-XH(I)
      DY=YH(I+1)-YH(I)
C
C        CHECK FOR ZEROS
C
      IF(ABS(DY).LT.1.E-10)GO TO 70
       IF(ABS(DX).LT.1.E-10)GO TO 40
C
C        GET THE SLOPE ,COSINE,AND SINE FOR EACH LINE(ROTATED 90
C        DEGREES FOR NORTH DIRECTED Y AXIS)
C
      S(I)=DY/DX
      T=1.5707963+ATAN2(DY,DX)
      CT(I)=COS(T)
      ST(I)=SIN(T)
      GO TO 60
   40 ST(I)=0.
      CT(I)=SIGN(1.,DY)
      S(I)=1.E+15
```

16. PROGRAM TO GENERATE SHIP'S CHARACTERISTICS (CHAR)

16.1 DESCRIPTION

Program CHAR provides a means for storing information describing the ship's physical characteristics and information pertaining to the vessel motion equations. (Refer to Table 2-1 for definitions of the 20 ship's characteristics provided by the program.)

16.2 PROGRAM CHAR LISTING

The program listing of CHAR, including the subroutine SIMQ for solving simultaneous linear equations, is presented in Tables 16-1 and 16-2 for the vessel safety model. This program is essentially unchanged from the original IEC[1] program SCHAG supplied by the United States Coast Guard under USCG Contract DOT-CG- 94638-A.

TABLE 16-1. PROGRAM LISTING FOR ROUTINE CHAR (CONT'D)

```
      VIS=VIS+VI(K)
      VI2S=VI2S+VI(K)**2.0
      VI3S=VI3S+VI(K)**3.0
      VI4S=VI4S+VI(K)**4.0
      VDS=VDS+VD(K)
      VIVDS=VIVDS+VI(K)*VD(K)
      VI2VDS=VI2VDS+VI(K)**2.0*VD(K)
   40 CONTINUE
C
C        STORE THE COEFFICIENTS FOR USE IN SIMQ
C
      A(1)=10.
      A(2)=VIS
      A(3)=VI2S
      A(4)=VIS
      A(5)=VI2S
      A(6)=VI3S
      A(7)=VI2S
      A(8)=VI3S
      A(9)=VI4S
      B(1)=VDS
      B(2)=VIVDS
      B(3)=VI2VDS
      N=3
C
C        GET THE COEFFICIENTS OF THE RESULTANT QUADRATIC
C
      CALL SIMQ(A,B,N,KS)
C
C        SAVE THESE IN THE ARRAY SC AND PUNCH AND PRINT
C        THE ARRAY.
C
      SC(18)=B(1)
      SC(19)=B(2)
      SC(20)=B(3)
      IF(IUNIT.EQ.0)GO TO 78
      WRITE(IUNIT,700)(SC(J),J=1,20)
   78 WRITE(6,701)(SC(J),J=1,20)
   77 CONTINUE
  700 FORMAT(5E15.7)
  701 FORMAT(1H1,(5E15.7))
      STOP
      END
```

TABLE 16-2. PROGRAM LISTING FOR SUBROUTINE SIMQ (CONT'D)

```
      JJ=-N
      DO 65 J=1,N
      JY=J+1
      JJ=JJ+N+1
      BIGA=0
      IT=JJ-J
      DO 30 I=J,N
C
C        SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN
C
      IJ=IT+I
      IF(ABS(BIGA)-ABS(A(IJ))) 20,30,30
   20 BIGA=A(IJ)
      IMAX=I
   30 CONTINUE
C
C        TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)
C
      IF(ABS(BIGA)-TOL) 35,35,40
   35 KS=1
      RETURN
C
C        INTERCHANGE ROWS IF NECESSARY
C
   40 I1=J+N*(J-2)
      IT=IMAX-J
      DO 50 K=J,N
      I1=I1+N
      I2=I1+IT
      SAVE=A(I1)
      A(I1)=A(I2)
      A(I2)=SAVE
C
C        DIVIDE EQUATION BY LEADING COEFFICIENT
C
   50 A(I1)=A(I1)/BIGA
      SAVE=B(IMAX)
      B(IMAX)=B(J)
      B(J)=SAVE/BIGA
C
C        ELIMINATE NEXT VARIABLE
C
      IF(J-N) 55,70,55
   55 IQS=N*(J-1)
      DO 65 IX=JY,N
      IXJ=IQS+IX
      IT=J-IX
      DO 60 JX=JY,N
      IXJX=N*(JX-1)+IX
      JJX=IXJX+IT
   60 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))
   65 B(IX)=B(IX)-(B(J)*A(IXJ))
C
C        BACK SOLUTION
C
```

17. MAIN PROGRAM PLOT

17.1 DESCRIPTION

a. This program plots the ship's tracks or the frequency array as specified by the user. The plotting routines used are the standard CALCOMP routines used by most systems. The plotting initialization may vary with each system. The subroutine PLOTS is used for initialization, and its calling sequence may vary.

b. Initially, the run number IRUN is read in. If it is greater than one the data tape, located on logical unit number 9, will have to be moved past the previous runs. In the harbor model at the end of each run the data value of NSHIPS (I was used in the main program) was set to zero. This zero is searched for and the run bypassed. Once the tape is positioned, the number of ships to be plotted is read in. If this value is zero, control is sent to statement 1000, where the frequency array of this run is plotted. Otherwise, the ship numbers to be plotted, their symbols, and the start and end times for the plot are read in. Next, the contour data are read in. This may be any one of the contours used by the main harbor model.

c. For the plot, the X-direction (North) will be plotted from right to left, and the Y-direction up and down on the plotting paper. Since the plotter will have a maximum of 10 inches in the Y-direction, the scaling equation is

X(SCALED) = (X(UNSCALED) * 10.) / RANGE

where RANGE = XMAX - XMIN

X is either coordinate. The coordinates of the origin are Y = -YMN*SCALED and X = -XMX*SCALED. The X-direction is opposite that required by the plotter, so the X-coordinate becomes X = XMX*SCALE. The pen is set at the origin, and is marked. Next, the contour and the islands are plotted.

TABLE 17-1. LIST OF VARIABLES FOR MAIN PROGRAM PLOT

ANG	The angle of the ship in degrees for use in subroutine SYMBOL
DATA	Data area used for plotting by subroutine PLOTS
DEG	Conversion factor for radians to degrees
DEL	The interval size for the frequency plot
END	The end time for plotting ships' positions
HD	The angle of the ship as read from tape. The coordinate system is with the X-axis North and the Y-axis East
IRUN	The run number for which plotting is to be done. Each run is separated by the value NSHIPS = 0
ISYM	Array containing the plotting symbols to be used for each ship plotted
ISP	Array containing the ship numbers to be plotted
NINT	Number of intervals to be used in the frequency plot
NIS	Number of islands in the contour
NPNTS	The number of points in the contour and each island
NSHIPS	The number of ships for which there are data on the tapes
NSPS	The number of ships to be plotted
OCCUR	Array of frequencies to be plotted against the distance 0 to RADI in NINT intervals
OUT	Array signifying which ships are considered out. These ships will not be plotted.

TABLE 17-2. PROGRAM LISTING FOR MAIN PROGRAM PLOT

```
C
C        ROUTINE FOR PLOTTING SHIP TRACKS OR FREQUENCY CURVE
C
      LOGICAL OUT(15)
      DIMENSION PX(15),PY(15),HD(15),SHR(15),ISP(15),ISYM(15),
     X NPNTS(15),XH(100,6),YH(100,6),DATA(1000)
      DIMENSION XX(102),YY(102)
      INTEGER OCCUR(102)
      DATA DEG/57.29578/
C
C        READ IN THE RUN NUMBER
C
      READ(5,100)IRUN
      IF(IRUN.LT.2)GO TO 70
C
C        IF NOT RUN NUMBER 1,SKIP RUNS ON TAPE TO CORRECT RUN NUMBER
C
      IRUN=IRUN-1
      DO 75 I=1,IRUN
C
C        READ IN THE NUMBER OF SHIPS TO BE PLOTTED
C
   77 READ(9)NSHIPS
C
C        IF NUMBER OF SHIPS EQUAL ZERO,GO TO FREQUENCY PLOT
C
      IF(NSHIPS.NE.0)GO TO 77
      READ(9)NINT
      READ(9)DUM
   75 CONTINUE
   70 CONTINUE
C
C        OTHERWISE,READ IN THE SHIP NUMBERS AND THE PLOTTING SYMBOLS
C
      READ(5,100)NSPS
      IF(NSPS.EQ.0)GO TO 1000
      READ(5,100)(ISP(I),I=1,NSPS)
      READ(5,150)(ISYM(I),I=1,NSPS)
C
C        READ START AND END TIMES FOR  PLOTTING
C
      READ(5,200) ST,END
      REWIND 9
C
C        READ THE NUMBER OF ISLANDS AND THE CONTOUR DATA
C
      READ(5,100)IDUM,NIS
      K=NIS+1
      READ(5,100)(NPNTS(J),J=1,K)
C
C        GET MAX AND MIN FOR SCALING
C
      XMX=-1.E30
      XMN=1.E30
      YMX=-1.E30
```

TABLE 17-2. PROGRAM LISTING FOR MAIN PROGRAM PLOT (CONT'D)

```
C
C         IF END TIME,GO TO END
C
      IF(TIME.GT.END)GO TO 500
C
C         READ THE DATA
C
      READ(9)(PX(I),PY(I),HD(I),SHR(I),OUT(I),I=1,NSHIPS)
      DO 40 I=1,NSPS
      K=ISP(I)
      IF(OUT(K))GO TO 40
      X=-PX(K)*SCALED
      Y=PY(K)*SCALED
C
C         X-DIRECTION IS HORIZONTAL AND TO LEFT OF ORIGIN
C
      ANG=90.-HD(K)*DEG
C
C         PLOT A SYMBOL AT THE SHIP'S POSITION
C
      CALL SYMBOL(X,Y,.05,ISYM(I),ANG,1)
   40 CONTINUE
      GO TO 1
C
C         SECTION FOR PLOTTING FREQUENCY ARRAY
C
 1000 CALL PLOTS(DATA,1000)
      CALL PLOT(0.,-11.,-3)
      CALL PLOT(0.,.5,-3)
C
C         SEARCH FOR ZERO
C
  700 READ(9)NSHIPS
      IF(NSHIPS.NE.0)GO TO 700
C
C         READ  THE DATA
C
      READ(9)NINT,RADI
      READ(9)(OCCUR(I),I=1,NINT)
C
C         MAKE ARRAYS FOR PLOT
C
      XX(1)=0.
      DEL=RADI/FLOAT(NINT)
      DO 710 I=1,NINT
      XX(I)=XX(I)+DEL
      YY(I)=FLOAT(OCCUR(I))
  710 CONTINUE
C
C         SCALE THE ARRAYS
C
      CALL SCALE(XX,10.,NINT,1)
      CALL SCALE(YY,8.,NINT,1)
C
C         DRAW THE AXIS
```

REFERENCES

1. IEC Technical Report 70-04, "All-Weather Harbor Navigation Model," International Engineering Company, Arlington, Virginia, August 1970.
2. "Handbook of Mathematical Functions," edited by M. Abramowitz and I.A. Stegun, Dover Publications, Inc., New York, 1965. Also available from National Bureau of Standards, Washington, DC, 1964 edition.