AD-A070 893 CASDE CORP TORRANCE CA F/G 5/2

PROGRAMMER'S MANUAL FOR COMPUTER PROGRAMS INVOLVED IN A STUDY O--ETC(U)

MAY 78 L B BROWN, R M DIJULIO, F J NICKELS DOT-CG-61276-A

UNCLASSIFIED USCG-D-9-78 NL

1 OF 2

AD A070893

Report No. CG-D-9-78

LEVEL II

10



# PROGRAMMER'S MANUAL FOR COMPUTER PROGRAMS INVOLVED IN A STUDY OF RESCUE BOAT PERFORMANCE FOR SELECTED COMMERCIAL VESSEL CASUALTIES

DEPARTMENT OF TRANSPORTATION • UNITED STATES OF AMERICA

MAY 1978

FINAL REPORT

Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151



Prepared for

**U.S. DEPARTMENT OF TRANSPORTATION**
**United States Coast Guard**
**Office of Research and Development**
**Washington, D.C. 20590**

## NOTICE

**Technical Report Documentation Page**

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| CG-D-9-78 | | 11 |

**4. Title and Subtitle**
6 Programmer's Manual for Computer Programs Involved in a Study of Rescue Boat Performance for Selected Commercial Vessel Casualties.

**5. Report Date**
May 1978

**6. Performing Organization Code**


**7. Author(s)**
10 L.B. Brown, R.M. DiJulio, F.J. Nickels, R. Saucedo

**8. Performing Organization Report No.**

**9. Performing Organization Name and Address**
CASDE CORPORATION,
P.O. Box 1291
Torrance, CA 90505


12 112p

**10. Work Unit No. (TRAIS)**
G-DSA-3309.01.02.1

**11. Contract or Grant No.**
15 DOT-CG-61276-A New

**12. Sponsoring Agency Name and Address**
Department of Transportation
U.S. Coast Guard
Office of Research and Development
Washington, D.C. 20590

**13. Type of Report and Period Covered**
9 Final Report. From November 1976 to May 1978

**14. Sponsoring Agency Code**
G-DSA-1/TP44

**15. Supplementary Notes**
The U.S. Coast Guard Office of Research and Development's technical representatives for the work performed herein were Mr. T.J. Sheehan and LCdr S.H. Davis.


**16. Abstract**
This volume presents the Programmer's Manual which describes the computer programs used to develop performance criteria for rescue boats involved in commercial vessel casualties. The casualties addressed are the Man Overboard and Abandon Ship for nine different types of casualties. Report No. CG-D-7-78 presents a detailed discussion of the overall study.

Both simulations, i.e., the Man Overboard Simulation and the Abandon Ship Simulation, are written in BASIC. Both simulations are intended to run on a micro or mini-computer system and were specifically developed for use on a Wang 2200 System.


18 USCG 19 D-9-78

**17. Key Words**
Performance Criteria, Man Overboard, Abandon Ship, Rescue Boat, BASIC, Computer Program

**18. Distribution Statement**
Document is available to the public through the National Technical Information Serivce, Springfield, Virginia, 22151

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 119 | |

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

New 411 267

# PREFACE

This report summarizes work conducted under Contract No.DOT-CG-61276-A by CASDE Corporation under the auspices of the U.S. Coast Guard, with Mr. T.J. Sheehan and LCdr. S.H. Davis serving as the Office of Research and Development's technical representatives for the work performed herein. The program manager was Dr. R. Saucedo. F.J. Nickels served as Naval Architect, R.M. DiJulio as Marine Engineer, and L.B. Brown as the System Analyst.

# 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 |

*1 in ≡ 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.


## 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 |

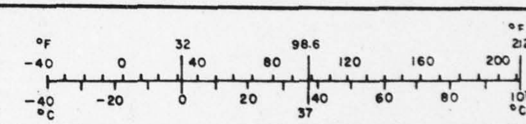

TABLE OF CONTENTS

# LIST OF APPENDICES

# I. INTRODUCTION

This manual describes the computer programs used to perform the Abandon Ship Computer Simulation Model (ASCSM) and the Man Overboard Computer Simulation Model (MOBCSM) simulations. The simulations run entirely on a micro (or mini) computer and were specifically designed for operation on a Wang 2200 System. The programs interactively acquire the information required to perform a simulation. For the ASCSM the microcomputer then performs the simulation and prints a report which tells the user the input conditions and the results of the simulations. For the MOBCSM the same result is achieved, but the interactive part of the sequence resides in a separate program which writes a disk file containing the input information for one or more simulation runs. The program that performs the simulation for the MOBCSM reads input information from the disk file, performs a simulation run, prints a report, and writes an out-put file. A third MOBCSM program reads the output file and produces another report; this may be useful when multiple copies of the same simulations are desired.

## 2. ABANDON SHIP SIMULATION

The Abandon Ship Simulation divides into three parts: the Operator Program, the Simulation Program, and the Output Program. The Operator Program provides an interactive input scenario to the operator asking for each piece of information needed to complete the data set required to perform a simulation. The specific questions appear in the User's Manual and in the program listing in Appendix A. The Simulation Program performs the number of simulations specified in the data set generated by the Operator Program. Each simulation data set contains values for variables that either remain constant for all the simulations or determine the parameters of random variables. After each simulation for a given data set the output data set is updated to form the basis for the output report on the simulations. The Output Program takes the output data set produced by the simulations and prints a neat looking report consisting of the input parameters set by the Operator Program and a statistical summary of the simulation results.

The Abandon Ship Simulation consists of a main program and a number of subroutines all written in BASIC*. The main program appears in Appendix A on page 1. Note that it consists mostly of calls to subroutines (e.g. GOSUB 1504). The main program calls the subroutine at 9000 to set the cursor on the video display to the upper left corner ("home") of the screen and erases the screen; it then displays the program title. The main program then asks for the serial number and initializes all variables and arrays. The subroutine called in line 1014, beginning in line 1504, and appearing on page A-2 uses the data statements on page A-3 from line 1372 to line 1440. In the main program line 1016 randomizes the random number generator by calling for, but not using, a number of random numbers. The number of random numbers thus skipped equals the integer part of the serial number supplied. The sub-

* BASIC is a registered trademark of Dartmouth University.

routines called in lines 1018, 1020, and 1022 constitute the Operator Program. The subroutine called in line 1024 is the part of the Output Program that prints the input data. Lines 1025 through 1040 constitute the Simulation Program while the subroutine called in line 1042 completes the Output Program; line 1044 goes back to start another run.

## 2.1 Operator Program

The Operator Program consists of three subroutines. The subroutine beginning with line 1046 handles the ship characteristics. The subroutine beginning with line 1102 gathers the rescue boat characteristics. Line 1134 begins the subroutine that takes care of the casualty type and the number of simulations desired for the run. These three subroutines appear on pages A-4 and A-5.

The subroutine that handles the ship type, beginning in line 1046, first "home"s the cursor and erases the screen (line 1048). The subroutine then puts "SHIP CHARACTERISTICS" followed by the menu of ship types on the video display. The response to the question "NUMBER?" (line 1058) is limited to the available ship type numbers by lines 1062 and 1063. If the user enters a number outside the range of numbers on the menu, the program asks the question "NUMBER"? again. After the user has entered a valid ship type number, the program moves the number of people on board (POB) the ship, the number and capacity of the lifeboats and liferafts from the ship characteristics array to array N5. The subroutine then computes the length of the lifeboat(s) and stores the length in array V1. Finally, the subroutine moves the probability that a rescue vessel is nearby to array P1.

Line 1102 begins the subroutine that requests, from the user, the rescue boat characteristics. The subroutine first sets the index I5 to zero (variable Z2) and erases the video display. The FOR-NEXT loop in lines 1110 to 1114 asks the questions and receives the answers. The actual questions are stored in array M5$ (a string array denoted by the terminal $) and processed in the subroutine that begins in line 1620. Both subroutines appear on page 4 of Appendix A. The subroutine beginning at line 1620 displays the question in array M5$ and puts the user reply into array D1. The subroutine then compares the response to a minimum value for the variable (in array M6) and a maximum value (in array M4) in line 1626. If the variable does not lie in this range, the subroutine asks the question again. Finally the subroutine increments the index for array D1 and returns to the calling subroutine at line 1114. The rescue boat length is computed and stored in array V1 in line 1118. In lines 1124 and 1126, the portions of probability $P_8$ which remain constant for all the simulations for the run are computed and stored in array V1. Finally the rescue boat capacities (crew and passengers) move to array N5 and the mean times to pick up one man and to perform a standard maneuver move to array T5. The text for the questions, the minimum allowed value and the maximum value appear in the data statements of lines 1426 through 1440, page A-3. The data statement in line 1424 contains one less than the number of questions.

The subroutine, beginning in line 1134 appearing on page A-5, to handle the casualty type and the number of simulations starts off like the other two discussed above by erasing the video display. The subroutine, lines 1138 through 1144, then displays "TYPE OF CASUALTY" followed by the menu of casualty types. As for the ship type, the program asks the user to choose the number, line 1146. The program checks the number, lines 1150 and 1151, and repeats the question if the number falls outside the allowed range. Lines 1154 through 1170 handle the override capability for the casualty. Next the subroutine picks probabilities P3 and P5 from the data supplied for the casualty type specified and puts the probabilities into array P1, lines 1172 and 1174. If the casualty is a collision, the subroutine sets probability $P_{14}$ to 0.98, lines 1175 through 1177. In any event a value for the index N3 is obtained from the casualty data, line 1178. The index N3 has a unique value for each primary casualty type (e.g. for a COLLISION &

& FIRE the primary casualty type is COLLISION and the secondary casualty type is FIRE). Finally the subroutine asks for the number of stimulations (lines 1180 through 1184 inititalizes the counters for the statistics, and returns (lines 1186 through 1198).

## 2.2 Simulation Program

The simulation program consists of a FOR-NEXT loop which calls three subroutines. The FOR-NEXT loop begins in line 1028 on page 1 of Appendix A and concludes with line 1040. The subroutines called in lines 1032, 1036, and 1039 together perform one simulation. These subroutines appear on pages A-6 through A-12. One subroutine handles each of the three sections into which the Abandon Ship Simulation naturally divides.

### 2.2.1 Set Up Casualty Conditions

The subroutine beginning at line 1674 performs the first section of a simulation. In this section of the simulation the number of people killed, knocked (or blown) into the water, and isolated by the casualty and the number of lifeboats/liferafts damaged are computed for the particular casualty being simulated. The type of casualty determines the mean value for these variables and the program selects a particular value from the appropriate statistical distribution using a random number generator. The random deviates are computed using the same equations described in reference (1). The casualties were divided into a primary casualty and a secondary casualty. The variables for the primary and secondary casualty are combined to satisfy the requirements of the model. For the primary casualty the following line numbers apply for each casualty type. For a fire, lines 1680 through 1688; for a collision, lines 1690 through 1712; for an explosion, 1714 through 1732; for a structural failure, 1734 through 1742; for a grounding or foundering, 1806 through 1816;

for a capsizing, 1750 through 1762. For the secondary casualty, the flooding is handled in lines 1766 through 1772 while for the fire lines 1774 through 1786 apply. No attempt will be made to explain this coding detail, but some of the features of this section will be noted and a typical portion explained in detail.

Since many of the variables computed at the beginning of the subroutine that performs the first section of the simulation use a distribution, the distribution computations appear in a subroutine. All the distributions appear on page A-12. Appendix B contains an annotated listing of all the variables used. The normal distribution begins with line 2218 (page A-12). The inputs to the normal distribution subroutine appear in array R1. The normal deviate (random variable selected from a normal distribution) appears in variable R6. Most of the normal distributions have only the mean specified. These distributions are bounded at zero on the low end. If the lower bound (of zero) is placed at the $2\sigma$ point of the distribution then a symetrical distribution requires an upper bound at twice the mean and a standard deviation, $\sigma$, of one-half of the mean. Since this condition occurs frequently, a subroutine that takes the mean (supplied in array element R1(1)) and computes the bounds and standard deviation then calls the normal distribution, appears starting at line 2258.

Now consider the case of an explosion and sinking. The primary casualty is an explosion which appears in lines 1714 through 1732. Line 1716 sets the mean for the special normal distribution to the mean number of people blown overboard by the explosion and goes to the subroutine to compute the normal deviate. Line 1718 rounds the normal deviate to the nearest integer and stores the value in the array element N5 corresponding to the number of people blown, knocked, or who jump overboard. See

Appendix B for the meaning of all variables. In a similar manner lines 1720 and 1722 compute the number of boats destroyed. Note that the total number of boats (lifeboats plus rescue boats) is multiplied by the fraction of boats destroyed by the explosion. This fraction is saved for later use in line 1724. In lines 1726 through 1730 the number of people isolated and killed are computed. For each of these the number of people on board is multiplied by the fraction affected for each case. The results get put in array N5. For an explosion, probability $P_5$ determines whether the secondary casualty is a fire or a sinking. Probability five ($P_5$) is only allowed two values, namely zero and one (0 or 1). In the subroutine, beginning at line 1134, that handles the casualty type, $P_5$ was put into P1(5). Line 1731, then, causes a branch based upon $P_5$. If $P_5$ equals one (1), control goes to line 1774; otherwise control continues at line 1768. The part of the subroutine beginning at line 1774 handles those casualties having a fire as the secondary casualty. This section also handles the casualty which consists of a fire only. Note that in this section, the various quantities generated add to those produced by the primary casualty, if any. Whether the secondary casualty is a fire or a sinking, line 1788 converts the fraction of boats destroyed into the fraction <u>not</u> destroyed. Line 1790 computes the total number of people in water while line 1792 sets the number of people in the rescue boat (exclusive of the rescue boat crew) to zero. Lines 1794 through 1800 perform the bookkeeping functions for the number of people in the water. X1(2) contains the number of people in the water for this simulation. Line 1796 calls to the middle the subroutine that begins at line 1656 (see page A-9) to update the values in Y1(1,1), Y1(1,2), and Y1(1,3). Y1(1,1) contains the sum of the number of people in the water for all the simulations in this run. Y1(1,2) maintains the minimum number of people in the water while Y1(1,3) becomes the maximum number of people in the water for any simulation of this run. Line 1800 jumps to line 1818 to compute the season, the wave height, the water temperature, and sea state and visibility functions. The season, lines 1818 through 1824, is a random number between zero and one from a uniform distribution. The subroutine beginning in line 2285 selects a random deviate from a uniform distribution having a minimum value supplied by R1(2) and a maximum value given in R1(3). The random deviate selected from a uniform distribution appears in V9. The wave height, lines 1826 through 1832, is a random deviate selected from a Rayleigh distribution. The mode supplied in R1(0) is linearly interpolated between the mid-summer value (season=0) and the mid-winter value (season=1). The subroutine that begins with line 2294 puts a random deviate selected from a Rayleigh distribution into R9. The water temperature, lines 1834 through 1846, is a random deviate selected from a normal distribution. The mean water temperature is linearly interpolated between a mid-summer and mid-winter value according to the season. The water temperature has a minimum value of 28 degrees, a maximum of 100 degrees and a standard deviation of 10 degrees. Line 1832 places the wave height in V1(2) for future use while line 1846 puts the water temperature in V1(1). Lines 1848 and 1850 compute the sea state function while lines 1852 through 1856 compute the visibility function multiplied by the sea state function. Line 1858 returns to the main program.

### 2.2.2 Get Boats Into Water

The subroutine beginning in line 1860 performs the second section of the simulation. In this section the boats and liferafts are launched. For davit launched boats (or rafts) consideration is given to whether the launch occurs from the high side or the low side. Lines 1862 and 1864 (see page A-8) set the number of rescue boats destroyed and the number launched to zero. Lines 1866 through 1888 contain a FOR-NEXT loop that determines whether a rescue boat was destroyed and if not whether or not the launch succeeds. The loop keeps track of the number launched and the number destroyed. Line 1890 tests to see whether any rescue boat(s) launch(es) succeeded. Line 1892 computes the number of people who must abandon ship in lifeboats and liferafts. Line 1894 computes the capacity of the rescue boat(s). Line 1896 adjusts the MTTP to reflect the number of rescue boats successfully launched. Line 1898 adjusts the inititial accumulated pick-up time to handle multiple rescue boats; note that if only one rescue boat will perform the pick-ups the initial accumulated time equals zero. Lines 1900 and 1902 compute how many lifeboats and liferafts have survived the casualty; while lines 1904 through 1920 determine how many get successfully launched. Line 1924 tests to see whether or not all the people can get into the lifeboats and liferafts. If so, lines 1926 through 1932 set the values of the excess capacity of the lifeboats and liferafts, the number of people in the rescue boat (excluding the crew), and jumps to line number 1940. If not all the people can get into the lifeboats and liferafts, the subroutine goes to line 1934 where the lifeboats and the liferafts are filled and the remainder attempt to abandon ship in the rescue boat(s). These computations appear in lines 1934 through 1942. If the number of people to abandon ship exceeds the total capacity of all the boats in the water, the subroutine goes to line 1958. Otherwise the subroutine continues to compute, in lines 1944 through 1954, the time for the casualty to develop, $T_C$, and the sum of the time to launch the rescue boat, $T_L$, and the time for the rescue boat crew to report to the rescue boat, $T_{RPT}$. The next line, 1956, returns to the main program. Lines 1958 through 1974 handle various situations which result in the rescue boat failing its primary mission. Appendix A page 9 lists various subroutines used in this section which are called from page A-8. In lines 1976 through 1992 the subroutine to compute the probability of a launch from the high side of a factor dependent upon the sea state (wave height). The computations use the short subroutine in lines 1994 through 2002 which limits the probability to the range between and including zero and one. Lines 2004 through 2014 compute the probability of a successful launch from the low side of the ship. Since the probability consists of a heel angle factor and the same sea state factor as for the high side launch probability, the subroutine requires less effort than the high side routine. The low side subroutine uses the subroutine beginning at line 1994 to limit the heel angle factor to the allowed range. The subroutine to compute the probability of a successful launch of a liferaft, which begins in line 2016, requires only three lines since there is no heel angle factor and the sea state factor has already been determined. The probability of a successful launch of the rescue boat depends upon the rescue boat characteristics, the heel angle of the ship, the sea state, and whether the boat is on the high or low side of the ship. The subroutine performing the necessary calculations appear in lines 2022 through 2042 on page A-9.

### 2.2.3 Rescue Men in Water

The program listing for the third major section of the Abandon Ship Simulation appears on page 10 and 11, of Appendix A. Page A-10 contains the main flow of this section while pages A-11 and A-12 lists the subroutines called from page A-10. In line 2048 on page A-10 a backwards FOR-NEXT loop is set up to pick up the people who have gotten into the water either from the primary casualty or who had to jump into the water because they were isolated from the lifeboats, rescue boat(s), and/or liferafts. The loop counts backwards, starting with the number of people in the water and counting down to one (1), to facilitate other computations within the loop. Lines 2050 and 2052 increment the total time to pick up and compute the survival time due to hyperthermia respectively. Lines 2054 and 2056 stops the simulation if the number of people remaining in the water equals zero; this can only occur if nobody was in the water to begin with. Line 2058 computes the probability that the man being picked up was knocked (or blown) into the water at the beginning of the primary casualty. Using the random number generator and the probability that the man was knocked into the water, the program decided, in line 2062, whether the man was knocked into the water or was isolated and jumped into the water. These two different ways of arriving in the water affect the length of time the man is in the water when the rescue boat arrives to rescue him and the probability that the man had a PFD when he went into the water. Lines 2064 through 2072 apply to the cases when the man was knocked into the water. Line 2066 selects the probability the man was wearing a PFD while line 2068 computes the time the man was in the water. Line 2070 decrements the number of men who were knocked into the water by one (1). Line 2072 jumps to line 2082 to complete the pick up processing. Lines 2074 through 2080 apply to the cases in which the man was isolated and had to jump into the water. Line 2076 selects the probability that the man was wearing a PFD while line 2078 computes the time the man was in the water. Note that both the probability (compare lines 2066 and 2076) and the time in the water (note lines 2068 and 2078) receive different values for the two cases. Line 2080 decrements the number of people who were isolated by one (1); this number really need not be decremented, but is done for completeness. From line 2082 through 2092 the subroutine completes the computation of the survival time and compares the survival time to the time in the water. Line 2084 calls the subroutine that begins on line 2270. The subroutine that begins with line 2270 appears on page A-12. This subroutine determines whether or not the man was wearing a PFD using the probability selected in either line 2066 or 2076 then computes the survival time due to floatation. Lines 2086 and 2090 complete the computation of the survival time for the man which is the lessor of the survival time due to floatation and the survival time due to hyperthermia. Line 2092 compares the survival time with the time in the water; if the survival time is greater than the time the man was in the water the control jumps to line 2096 where the saved statistics get accumulated. Otherwise the man is lost and control jumps to line 2102 via line 2094. The lines from 2096 through 2100 both increment the number of saved (line 2098 which calls the subroutine beginning in line 2210) and the number of survivors in the rescue boat (line 2100). Lines 2102 through 2150 complete the FOR-NEXT loop. Line 2102 tests to see whether or not the rescue boat is full. If not, the control jumps to the end of the loop in line 2150. Otherwise, the routine attempts to transfer all or part of the men in the rescue boat to a lifeboat or liferaft. Lines 2104 through 2118 establish the time required to transfer one man from the rescue boat to a lifeboat or liferaft; the time has a normal distribution about the mean value supplied as one of the rescue boat characteristics. Line 2120 determines whether or not a rescue vessel is nearby by using the random number generator and the probability that a rescue vessel is nearby. If a rescue vessel is nearby, control goes to line 2142 where the people rescued by the rescue boat are transferred. Otherwise, control proceeds to line 2122 where the number of people in the rescue boat gets compared with the remaining capacity available in the lifeboats and liferafts. If there remains sufficient capacity for all the survivors in the rescue boat, control shifts to line 2142. Note that this is the same place where control went if a rescue vessel was nearby. The routine in lines 2142 through 2148 simply reduces the number of people in the rescue boat to zero, increases the pick up time by the time to transfer one man times the number of men transferred, and reduces the available capacity in the lifeboats and liferafts by the number of people transferred. If the available capacity of the lifeboats and liferafts is less than the number of people in the rescue boat, control proceeds to line 2124. Lines 2124 through 2130 decrease the number of people in the rescue boat by the available capacity of the lifeboats and liferafts, sets the available capacity to zero, and increments the time to pick up by the time to transfer one man times the number transferred. Line 2132 tests to see if the number of people in the rescue boat is less than the rescue boat capacity. If so, control jumps to line 2150 which ends the FOR-NEXT loop. Otherwise, control proceeds to line 2134 where the tally of total number of people not rescued by the rescue boat is increased by the number of people remaining in the water. This is accomplished by lines 2136 and 2138. Note that line 2138 calls the subroutine that begins at line 1656. This subroutine, which appears on page A-9, updates the cumulative tallies for any specified variable. In line 2140 control jumps to line 2152; note that this constitutes an abnormal exit from the FOR-NEXT loop. Since all the boats have been filled to capacity, the simulation terminates after performing the necessary bookkeeping to update the various cumulative tallies. Lines 2142 through 2148 were discussed above while line 2150 completes the FOR-NEXT loop. Lines 2152 through 2162 compute the ratio of number saved during the simulation to the number in the water, updates the cumulative tallies for the ratio and total number saved and returns control to the main program. Lines 2174 through 2208 do the actual computations required to update the cumulative tallies for the saved ratio per simulation and the total number saved. Lines 2210 through 2216 constitute the trivial subroutine used to keep track of the number saved during each simulation.

## 2.3 Output Program

The output program has two parts. The first part prints out the input parameters established by the user before the simulations begin. The second part prints a report summarizing the results of the simulations for the run. The subroutine that performs the first part begins with line 1200 and appears on page 13 of Appendix A. The subroutine consists mostly of PRINT statements.

All the verbage that appears on the printed report appears in the PRINT statements except the heading for the report and the string "MEAN TIME TO". The statement that PRINTs the heading appears in line 1206; the string variable, A1$, receives the string "ABANDON SHIP SIMULATION" at the very beginning of the program (see page A-1 line 1004). Line 1220 sets up the string M7$ to "MEAN TIME TO". The statement in line 1202 causes the PRINT statements to output to the printer rather than the video display. Most of the print statements contain a reference to the TAB function. This function simply inserts the number of spaces specified by the argument. Line 1252 tests the casualty type and skips the line that prints the length of colision damage if the casualty does not involve a collision. Line 1256 skips the PRINT statement that prints the heel angle for casualties that do not involve a sinking. The statement in line 1262 causes subsequent PRINT statements to output to the video display.

The subroutine beginning with line 1266, appearing on page A-14, prints the summary report of the results of the simulations. Line 1268 selects the printer for statement output. Because the FOR-NEXT loop that counts the number of simulations starts at zero, the variable, X1(1), which stops the run has a value of one less the number of simulations. Line 1269 sets N equal to the number of simulations. Lines 1271 through 1276 compute and print the overall fraction of men in the water rescued by the rescue boat. If the total number of people in the water for all the simulations in the run equals zero (0), line 1271 sets V equal to minus one (-1) and transfers control to line 1274. Otherwise control proceeds to line 1273 where the overall fraction of men in the water rescued by the rescue boat gets computed. Note that the array element Y1(1,1) contains the total number of people in the water for all the simulations in the run while array element Y1(2,1) contains the total number of people rescued by the rescue boat for all the simulations. Line 1273 rounds the variable V to three decimal places; this will force the un-formatted print statement to output the fraction with no more than three decimal places. Lines 1274 and 1276 sets up four variables to provide four "tab stops" for the subsequent print statements and print the output line "OVERALL FRACTION OF **********************". Line 1277 prints the text "MEN IN WATER RESCUED". Line 1280 tests the variable V. If V is less than zero, which can only occur if the total number of people in the water equals zero (see lines 1272), line 1281 prints three dashes (---) and goes to line 1283. Otherwise, line 1282 prints the variable V. Lines 1283 and 1284 print the line "BY RESCUE BOAT *********". Lines 1285 through 1287 print the headings on the columns to follow and sets the temporary variable M equal to N, the number of simulations. Lines 1288 through 1306 print the first three lines following the column headings. Line 1288 begins a FOR-NEXT loop to print the three lines. Line 1292 reads a data statement; the first time around the loop the data statement in line 1484 appearing on page A-15 gets read. Subsequent times around the loop will read the data statements in lines 1486 and 1488. Variable M equals the number of simulations the first time around the loop and the number of times the rescue boat got deployed the second and third times around the loop. The value of M gets printed in the column headed "NO.TIMES". Next the entries in the columns headed "MEAN", "MIN.", and "MAX." require printing to complete the line. The mean requires dividing by the number printed in the column "NO. TIMES"; this number can be zero. Therefore, line 1294 tests the value of M and skips the computation if the mean for M equals zero. Lines 1296 and 1297 set the total number for the mean to zero if the total turned up negative. Line 1298 computes the mean, and puts it in temporary variable Q1. Line 1300 puts the minimum into variable Q2 while line 1302 sets Q3 to the maximum. Line 1304 sets the temporary variable M equal to W. The variable W accumulates the total number of times the rescue boat got deployed during all the simulations in the run(see line 2046 page A-10). Line 1306 completes the FOR-NEXT loop that began with line 1288. Lines 1310 through 1330 constitute another FOR-NEXT loop. This loop would be almost identical to the loop discussed above if all the remaining lines required only one line of text and had all the columns printed. The first two lines printed by this loop require printed entries only in the column headed "NO. TIMES". Line 1311 sets variables Q4, Q5, Q6, and Q7 to zero. Line 1312 prints one (1) space,reads the first line of text, and, for the first two lines only, jumps to line 1319. Line 1313 prints the first line of text, reads the second line of text, and either prints the second line of text or jumps to line 1319. Line 1313 jumps to line 1319 for the third time around the loop. For the fourth, and last time around the loop control can arrive at line 1314 which reads the third line of text. In any event control reaches line 1319, for each time around the loop, where variable Q4 gets set to Y1(I,1), the"NUMBER OF TIMES". For the first two times around the loop lines 1320 and 1321 print the last line of text, the "NUMBER OF TIMES", and zero for the other variables, then skip to the end of the loop. For the third and fourth times around the loop control goes to line 1322 which tests the"NUMBER OF TIMES". If greater than zero, control goes to line 1324 to compute the mean (variable Q5). Otherwise control goes to line 1323 to set Q5 to zero and jump to line 1326. Lines 1326 and 1328 set variables Q6 and Q7 to the minimum and maximum respectively. Line 1329 prints the last line of text and the variables Q4, Q5, Q6, and Q7. Note that the print statements are executed by calls to subroutines. These subroutines appear on page A-15. Line 1330 completes the FOR-NEXT loop and line 1332 completes the subroutine. Note that the SELECT statement in line 1332 causes subsequent print statements to output to the video display.

## 3. MAN OVERBOARD SIMULATION

The Man Overboard Simulation also consists of three programs all of which are written in BASIC. The first program interacts with the user to obtain the information required to make one or more simulation runs. This first program, the Man Overboard Operator Program, provides an interactive input scenario to the operator asking for each piece of information required to complete the data set needed to perform a simulation run. The Operator Program prints a summary of the information required by the second program. The second program, the Man Overboard Simulation Program, gets its input from the disk file written by the first program, performs the simulation(s) specified, produces the reports requested by the user, and writes a disk file containing the input information and the results of the simulation(s) performed. The user may request five different kinds of print-outs from the Simulation Program. Three of these print-outs served as aids in program development, debugging, and check-out and consequently would not be required by the general user. The other two kinds of print-outs contain the results of one or more simulation runs. One print-out produces five lines of print for each simulation containing the results of the simulation. This kind of print-out goes by the name "COMPRESSED PRINT". The remaining kind of print-out, called "REGULAR PRINT", produces a one page report for each simulation run. The report shows the variable input data and the results of the simulation. The third program, the Output Program, reads the disk file created by the second program and produces a report identical to the "REGULAR PRINT" report produced by the second program. The third program is used to obtain regular (one page per simulation) print from one or more simulation runs performed with only the compressed print requested. Normal usage would request either regular or compressed print but not both.

### 3.1 Man Overboard Operator Program

The Man Overboard Operator Program consists of a main program, five subroutines and a number of DATA statements. The main program divides into four parts: Initialization, Ship Characteristics, Rescue Boat Characteristics, and Conclusion. A listing of the program appears in Appendix C.

#### 3.1.1 Initialization

A listing of the main program appears in Appendix C one pages 1 and 2. Appendix D contains an annotated list of the variables and arrays used in the program. The main program, in line 10, first calls the subroutine at line 9200 to home and erase the video display. In lines 22 through 28 the main program DIMensions the arrays required by the remainder of the program. Lines 30 through 34 request the beginning and ending serial numbers from the user. Lines 40 and 42 initializes a disk file to receive the information entered by the operator while lines 68 begins a FOR-NEXT loop to handle the number of simulation runs implied by the beginning and ending serial numbers entered by the user in lines 30 through 34. Lines 70 through 76 print "INITIALIZING" on the video display and call subroutines at 9000 and 9100 to initialize the constants and arrays. When the initialization completes, line 78 calls subroutine 9200 to home and erase the video display removing the message "INITIALIZING" placed on the screen at line 72.

Subroutine 9000, which appears on page C-3, assigns values to a number of variables that do not change during the Operator Program. Many of these variables no longer have any use in the Operator Program. During program development the Man Overboard Simulation consisted of one large program. All these assignments were required before the Operator Program was separated from the Man Overboard Simulation Program.

Page C-3 has a listing of subroutine 9100 which assigns values to the elements of array variables. The values for these variables appear in the DATA statements in line numbers 9800 through 9994 on page C-4. Lines 9100 through 9120 assign values to the elements of the string array M1$ and the numerical arrays M6 and M7. The values for these variables appear in the DATA statements in lines 9800 through 9885. Note that each line, containing one DATA statement, consists of a string for array M1$ and a value for each of the arrays M6 and M7. Lines 9125 and 9150 assign values to the string array T1$ and the numerical array T1. Note that the array T1 has two subscripts. The first subscript specifies the ship number while the second subscript specifies the characteristic number. The strings for array T1$ and the numerical values for array T1 appear in the DATA statements in lines 9900 through 9980. Note that each line contains the string for one element in array T1$ and all the characteristics that pertain to that ship type. Finally the code in lines 9160 through 9180 reads the numerical variables in the DATA statements of lines 9990 through 9994 into array P2. The statement in line 9199 returns control to the main program.

The subroutine beginning in line 9200 simply homes and erases the video display then returns control to the calling program.

### 3.1.2 Ship Characteristics

Lines 80 through 250 perform the second function of the Operator Program by setting the ship characteristics. Lines 80 through 110 put the messages"SHIP CHARACTERISTICS" and "TYPE OF SHIP?" followed by a menu of ship types on the video display. Line 120 displays the message "NUMBER OF SHIP DESIRED" and receives the reply while line 130 tests the response to insure the user typed a valid number. With a valid response, the program proceeds to line 140 to clear the screen in preparation for the next question. The subroutine at

9300 puts the question "WANT STANDARD SHIP?" on the screen. Subroutine 9300 appears on page C-4. The variable J1 determines which question will display. Variable M3 obtained its value when subroutine 9000 was called in line 72. Line 160 tests the response to the question and transfers to line 240 if the user supplied an affirmative reply. Lines 170 through 230 handle the cases in which the user elected not to use the standard ship values. Lines 180 through 230 allow the user to change the ship length, speed, and mean heading error. The program, for each value, presents the standard value and requests the desired value. The user can, of coarse, enter the standard value for any or all the values that the program allows the user to change. Line 240 concludes the section on ship characteristics. In line 240, which the program reaches whether or not the user selected a standard ship, array element P2(10) receives a value. P2(10) is the probability the ship will recover the MOB directly which depends upon ship type.

Subroutine 9300, which is called three times by the main program, appears on page C-4. The subroutine displays a question on the video display, receives the user reply to the question, and tests the reply. If the reply lies outside the allowable range of values precribed for the variable, the subroutine starts over. Line 9310 displays the J1th string in the string array M1$. M6 is an array of minimum values while M7 contains the maximum values allowed for each variable inputted by the user. During the initialization, subroutine 9100 sets up these arrays (see lines 9100 through 9120). For values that require a "YES" or "NO" response, the minimum and maximum have the same value (ZERO); this serves as a flag which the subroutine tests in line 9315. If the input variable requires a "YES" or "NO" response,the program transfer to line 9370. Line 9320 receives the user response to the question. Note that the INPUT statement in line 9320 supplies the question mark (?) for the question. In lines 9325 and 9326 the user response sets compared to the minimum and maximum allowed for the variable. For user responses outside the allowed range, the control returns to line 9310 to repeat the question. Otherwise control proceeds to line 9340 to increment the index to the array which receives the user replies. The index to the question (array M1$), minimum (array M6), and maximum (array M7) arrays, J1, receives its values from the calling program (see lines 150, 180-230, and 290-310). In line 9360 control returns to the calling program. For variables that require a "YES" or "NO" response, line 9315 transferred control to line 9370 which starts a new line and prints the remainder of the question thereby defining the acceptable answer. Line 9375 receives the user response. Lines 9380 and 9385 test the response. For an incorrect response, control returns to line 9310 to display the question again. Note that while the program requests a "1" for a "YES" response and a "0" for a "NO" response, by changing lines 9370 through 9385, any other response, e.g. "Y" or "N", will become the acceptable responses for all "YES"-"NO" questions except the question "WANT TO INPUT NEXT RUN?". For "YES"-"NO" questions now under discussion, line 9399 returns control to the calling program.

### 3.1.3 Rescue Boat Characteristics

The third section of the main program receives the rescue boat characteristics. The main program coding appears in lines 250 through 365. Since the object of the man overboard simulation involves comparing the performance of different rescue boats, all the variables relating to the rescue boat appear as user supplied variables. This allows maximum flexibility precisely where the user needs it. Line 250 resets the index to the "reply array" to unity while lines 260 through 280 clear the video display and display the message "RESCUE BOAT CHARACTERISTICS". Lines 290 through 310 ask the questions and receive the replies for each of the rescue boat variables up through the first question that requires a "YES"-"NO" answer. Note that subroutine 9300 does most of the work. Lines 320 through 335 process the reply to the last question by setting the value of the "reply array", D1, to zero for a "NO" response and unity for a "YES" reply. Lines 340 through 355 handle the last rescue boat question and response. A "NO" response to this question, "SEARCH WITH RESCUE BOAT ONLY", allows the program to use a value for the probability the ship is restricted from turning that depends upon the type of ship. A "YES" response forces the probability to unity. Finally, lines 360 and 365 set up the probabilities P5, P6, and P15 in array P2.

### 3.1.4 Conclusion of the Operator Program

This section discusses the end of the Operator Program. Lines 370 through 460 obtain the number of simulations desired and compute those variables that remain constant during a simulation run. Lines 370 through 378 ask the question "NUMBER OF SIMULATIONS?" and receive the reply. Lines 380 through 460 compute the variables that remain constant during a simulation run (which usually consists of many simulations having the same input data). All these variables are stored in array T5. The annotated list of variables in Appendix D describes each element in array T5. Lines 500 through 660 write the input data on the disk. Line 680 calls subroutine 8600 which prints the data the user has specified either directly or implicitly. Subroutine 8600 also prints some of the variables which remain constant throughout a simulation run. The details of this subroutine appear in the following paragraph. In line 690 the program tests to see if the current serial number is the last serial number requested by the user at the beginning of the program (line 34). If the current serial number is the last, control jumps to line 799. Otherwise the program proceeds to line 700. Lines 700 through 750 ask the user whether or not to gather the information for the next simulation run which is identified by serial number (line 720). If the user elects not to proceed, the program jumps to line 800. Otherwise control continues to line 760 which RESTOREs the pointer to the DATA statements. Line 799 concludes the FOR-NEXT loop that began in line 68. In any event,eventually control reaches line 800 which prints the message "INPUT PHASE COMPLETE" on the video display. Line 820 closes the disk file into which the program wrote the input data and terminates with the STOP statement.

Subroutine 8600 bears the REMark "DEBUG PRINTOUTS". This comment remains from the time when the entire Man Overboard Simulation consisted of one large program. The program was too large, however, to run in the memory available at development time. At that time the printout discussed in this paragraph served as a debugging aid to verify that the input part of the program functioned properly. Now, however, with the Operator Program separate from the Simulation Program the printout serves the necessary function of identifying the output with the input. Line 8602 selects the line printer for output. Line 8610 prints the ship type while lines 8615 through 8635 print the ship characteristics. For the definition of the elements in the T1 array see Appendix D. Note that in the Operator Program the T1 array has two indexes. In the other Man Overboard Programs, the T1 array has only one DIMension (index). In the Operator Program the ship type the user selects determines the first index. The Operator Program stores, on disk, the characteristics for the ship selected by the user. The other Man Overboard Programs do not need the characteristics of the other ships available to the user of the Operator Program. Note that the Operator Program prints two lines containing the characteristics of the ship selected by the user with six characteristics on each line. Line 8622 tests the index into the characteristics. Line 8624 starts a new line for characteristic number seven. Line 8640, prints the heading "RESCUE BOAT" preceeded by a blank line. Lines 8645 through 8658 print the rescue boat characteristics entered by the user in two lines with six characteristics per line. Lines 8562 and 8563 serve to start the second line with the seventh characteristic just as lines 8624 and 8626 did for the ship characteristics.

In a similar manner line 8660 prints the heading "PROB." preceeded by a blank line while lines 8662 through 8669 print the array of probabilities in array P2. Note the probabilities appear on three lines since there are fifteen probabilities in the array. Lines 8665 through 8667 cause new lines to begin for the seventh and thirteenth probabilities. Finally lines 8670 through 8694 print the computed values that remain constant for all runs in a simulation run. In this case each of the variables printed has an identification on the line above it. Line 8696 selects the video display for future "printed" output. Line 8799 returns to the main program.

### 3.2 Man Overboard Simulation Program

The second of three programs that make up the complete Man Overboard Simulation performs the simulation. The first program, the Man Overboard Input Program, writes a disk file that serves as input to the second program. The second program performs the simulation(s) requested and writes a disk file that serves as an input to the third program. The second program, under user control, can produce the same report as produced by the third program. A listing of the second program appears in Appendix E. The second program, Man Overboard Simulation Program, consists of a main program, and a number of subroutines. The main program divides into four parts: Input Section, Simulation Program, Output Section, and Subroutines. The Input Section extends from the beginning to line 285. The Simulation Section begins with 300 and extends to line 2570. The Output Section appears in lines 5900 through 6099 and the Subroutines encompass the remainder of the program.

#### 3.2.1 Input Section

The Input Section which appears on page E-1 calls subroutines on pages E-14 and E-15 and jumps to a line that appears on page E-2. The first executable line of the program, line 10, calls subroutine 9200 on page E-15 to home and erase the video display. Next, in lines 20 through 28, the program prints the message "INITIALIZING" and DIMensions the arrays used by the remainder of the program. The subroutine at 9000 gets called to initialize the variables that remain constant throughout all the simulation runs. In lines 40 through 69, the program interacts with the user to determine which print-outs to produce during the simulation runs to follow. Note there are three print-outs to aid in debugging the program. The remaining two print-outs occur for each simulation run if requested by the user. The regular print-out produces a one page report for each simulation run while the compressed print-out produces five lines for each simulation run. Line 90 requests, from the user, the name of the disk file containing the input data for the set of simulation runs and the name of the file on which to write the output data. Lines 92 through 98 produce the heading for the compressed print-out if the user elected to have a compressed print-out. Otherwise, lines 92 through 98 have no effect. Lines 100 through 262 read the input data for one simulation run from the disk and write a copy of the data to the output disk file. Lines 118 and 250 and 152 through 156 work together to open the output file for the first input data set only. Finally, lines 266 through 285 produce debug print-out #3 if requested by the user; otherwise these lines have no effect.

The above discussion skipped lightly over the statement that results in a jump off the page. After a simulation run, see line 6099, control returns to line 140 to read the next data set from the input disk file. If the last data set has already been read, the program will, then, at this point, read an end-of-file. Line 130 causes control to go to line 400, see page E-2, for an end-of-file. Lines 440 through 462, on page E-2, formerly received jumps from page E-1, but are no longer used.

### 3.2.2 Simulation Section

The Simulation Section includes a large fraction of the total Simulation Program. The Simulation begins at line 300 on page E-2. Lines 300 through 360 set the elements of the arrays used to accumulate the statistics for a simulation to the correct initial values. Line 399 jumps around the lines that handles the end-of-file condition. The simulation then continues at line 470 to begin the loop which performs the requested number of simulations. Line 480 contains the FOR statement. The corresponding NEXT statement appears in line 5999 on page E-8. Lines 490 through 499 reset the flags that must be reset for each simulation. Lines 500 through 784 compute all the variables that remain constant throughout a given simulation. Most of these variables depend upon random numbers. Lines 500 through 784 appear on pages E-2 and E-3. Lines 500 through 625 compute the constant term of the navigational heading error. Note that this constant term has a normal distribution. Line 610 calls subroutine 8000 to obtain the normal variant. Subroutine 8000 appears on page E-13 and is discussed below on pages 40 and 45. Next lines 630 through 650 set the season or time of year for the simulation. Since the season has a uniform distribution with a range from zero to unity which exactly matches the distribution and range of the random number generator, the code does not require the uniform distribution subroutine. Lines 660 through 668 compute the environmental visibility using the Rayleigh distribution subroutine at line 8200 which appears on page E-12. Pages 45 & 46 below contain a discussion of subroutine 8200. Lines 670 through 676 use subroutine 9500 to compute the distance of visibility from the ship and the rescue boat; the subroutine discussed on pages 48 and 49 below appears on page E-15. Lines 680 through 698 compute the water temperature using subroutine 8000 to obtain a normal variant. Line 690 linearly interpolates, by season (or time of year), between a mid-summer mean, T1(9), and a mid-winter mean, T1(10), to obtain the mean temperature. Line 700 through 730 use the subroutine at 8400, appearing on page E-14 and discussed below on page (46), to compute the survival time due to hypothermia. Next the program computes the significant height of the waves in lines 740 through 754, page E-3. The significant height of the waves has a Rayleigh distribution with the mode determined by the formula in line 750. The formula linearly interpolates between a mid-summer mode of significant wave height and mid-winter value according to the season. The probability of a successfull recovery of the rescue boat depends upon the number of falls (1 or 2) and the significant height of the waves. Lines 760 through 766 compute the probability of successful recovery of the rescue boat using the formula in line 766. The Constants P3, P4, P5, and P6 receive values in subroutine 9000 as discussed above on page (25) and below on page (47) and appearing on page E-14 and E-15. Lines 770 through 784 compute the last of the variables that remain constant throughout a simulation. Lines 776 through 782 compute the rescue boat speed and restrict the minimum value to three (3) knots. The boat speed depends upon the specified boat speed in calm water, the boat speed in eight foot (8') waves, and the significant height of the waves. Line 784 computes the ratio of the ship speed divided by the boat speed. Next, if the user requested debug print #1, the program prints the variables that remain constant through a given simulation run. This print-out resides in subroutine 8700, discussed on pages 46 and 47 below listed on page E-14, gets called in line 790.

Most of the remainder of the simulation consists of five parallel paths of which, for any given simulation, only one will be followed. The five paths consist of two paths when the Man Overboard (MOB) was seen going overboard and three paths when the MOB was not seen going overboard. For the case in which the MOB was seen going overboard one path handles the situation in which the captain elects to turn and perform the search. The other "seen" path, which applies when the ship is restricted from turning and uses the rescue boat for the search. The paths relating to cases in which the MOB was not seen going overboard handle search with ship, search with rescue boat, and search with both rescue boat and ship. Which path the program will follow depends upon random variables and user supplied parameters. In general, in a simulation run, each of the five paths will be traversed during at least one simulation. With one exception lines 840 through 1520 handle the paths for which the MOB was not seen going overboard. That exception involves the path in which the ship is restricted from turning. For this exception the program uses the code for the parallel case in which the MOB was seen going overboard. With the exception noted above, lines 1600 through 2085 handle the paths covering the cases in which the MOB was seen going overboard. Also there are subroutines that get called at different points along each of the paths. One of these subroutines, see lines 6500 through 6599, page E-11, gathers the statistics when called. The variable J9 and array element F(1) together determine which set of statistics to up date. Separate sets of statistics accumulate for the "seen" and "not seen" cases; five sets for each. For each set of statistics, subroutine 6500 accumulates the number of times the event occurred, the sum of the times at which the event occurred (to compute the mean time), the minimum time, and the maximum time. Two other subroutines may get called in each path. One of these, lines 7000 through 7599, pages E-11 and E-12, sets up array T3 with values needed to perform a search. The other subroutine, lines 9400 through 9499, page E-15, performs the search for the MOB. Note that many conditions can cause the rescue attempt to fail. Almost every failure results in exit from the main program flow to a unique line. This allows each cause of failure to set a flag indicating what caused the attempted rescue to fail. In addition as the program proceeds down a given path flags are set at a strategic junctures in the path so the course of each simulation can be traced precisely if the user selects debug print #2 which prints the values of the flags after each simulation.

### 3.2.2.1 MOB Seen or Not Seen Going Overboard

Picking up the detailed discussion again at line 800, page E-3, the next executable statement, line 830, uses the random number generator and the probability the MOB was seen going overboard, P1 in P2(1), to perform the "seen"-"not seen" dichotomy. The control jumps to line 1600 if the MOB was seen going overboard. Otherwise control continues to line 840. Line 850 sets the first element in the flag array, F, to unity. F(1) equal to one indicates "not seen" while F(1) equal to two indicates "seen". Line 855 increments the tally of the number of times the MOB was not seen going overboard, S8(2). Note subroutine 6500 was not used because this event requires only a count, the time at which the event occurs has no significance. Line 860 sets the variable P to the probability the MOB was wearing a PFD. Note that this probability depends upon whether or not the MOB was seen going overboard. Line 870, calls subroutine 8500, page E-14, to compute the survival time due to drowning which depends upon whether or not the MOB had floation of some kind (PFD or Life Ring). Lines 880 and 882 set the survival time to the lessor of the time to drown and the time to die due to hypothermia. Note that the time to drown used a variant from the Rayleigh distribution. All the distribution subroutines return the variant in variable R3. Next, lines 888 through 920 compute the time since the MOB was last seen and the time from when the MOB went overboard until missed. The time since last seen, in the simulation, has a minimum of 15 minutes, a mode of 45 minutes and a Rayleigh distribution. Since the Rayleigh distribution subroutine at line 8200 has a minimum of zero, lines 890 and 900 select a variant from a Rayleigh distribution having a 30 minute mode and line 910 adds the 15 minute minimum to the variant to produce the desired time since last seen. The time from when the MOB went overboard until missed consists of a uniformly distributed fraction of time since last seen. This computation occurs in line 920. Note that this method of computing the time missed can result in the time missed less than the minimum time last seen. Lines 922 through 928 complete the computation of the navigational error by adding the term that depends upon the time period from the time the MOB went overboard until the crew missed the MOB. The running time for the simulation appears in the variable T which line 930 sets to the time to miss the MOB. The survival time gets put into variable TO in line 934 while line 940 tests to see if the MOB still lives. If the MOB expires at or before getting missed, control transfers to line 3100, page E-7, which sets flag F(6) to unity and jumps to the common EXIT point at line 5900, page E-8. Lines 945 through 990 compute the time to notify the bridge, a normal variant, increment the running time, T, and test to see if the MOB still lives. If the MOB has expired control, in line 990, goes to line 3200 on page E-7 where flag F(6) gets set to two and control passes to line 5900 on page E-8. If the MOB lives, control proceeds normally to line 1000, page E-4, to gather statistics for the MOB's who live until the bridge receives notice of a MOB situation. Note that subroutine 6500, page E-11, handles the updating of approriate tallies.

3.2.2.2 Ship Restricted or Not Restricted From Turning

Line 1030, page E-4, performs the "ship restricted from turning"-"not restricted" dichotomy. Control proceeds normally to line 1040 for the "not restricted" case and jumps to line 1500 for the "ship restricted from turning" case. In line 1050 flag F(2) receives a unity value to indicate "ship not restricted".

3.2.2.3 Search With Ship When MOB Not Seen

Line 1060 performs the final branch determining which of the five paths the simulation will follow. In this case the user can, in the Operator Program, elect to use or not to use the rescue boat for the search by the reply given to the question "USE BOAT FOR SEARCH?". If the user has elected to use the rescue boat for search control jumps to line 1200 to conduct the search with both the rescue boat and the ship. Otherwise control proceeds normally to line 1100. Thus when control reaches line 1100 the program lies on the path for "MOB not seen", "ship not restricted", and "search with ship only"; line 1110 sets flag F(3) to unity to indicate "search with ship only". The variable Z, which serves as a switch variable in subroutine 7000, pages E-11 and E-12, gets set to three in line 1120, page E-4. Line 1130 calls subroutine 7000 to set up array T3 with values approriate to the type of search required. Line 1140 calls subroutine 9400, page E-15, to perform the search. Line 1148 increments the running time, T by the time for the ship or rescue boat to return to the MOB's position. Line 1160 tests the results of the search. If the search failed line 1160 causes a jump to line 3300, on page E-7. Note that the search routine, line 9400 on page E-15, handles all the simulated events, except rescue boat launch, from the time the bridge receives notification until the MOB is found alive or expires. Therefore, if the search fails, the main program must determine whether or not the man expired before the ship or rescue boat returned to the position at which the MOB went overboard. This distinction results from the gathering of statistics at the point when the ship or rescue boat returns to the position at which the MOB went overboard. Lines 3300 through 3399 first set flag F(6) to three then jump to the common exit at line 5900 if the MOB expired before the ship or rescue boat reached the MOB's position. If the MOB remained alive at the time the ship or rescue boat returned, line 3340 gathers the statistics by calling subroutine 6500 on page E-11 then jumps to the common exit point at line 5900, page E-8. On the other hand, a successful search allows control to continue normally to line 1170. Lines 1170 through 1185 increment the running time, T, by the time to search and find the MOB and gather the statistics for the time at which the MOB's position was reached by the ship or the rescue boat and the time at which the MOB was found. Then line 1190 jumps to line 1400, page E-5, to complete the rescue.

### 3.2.2.4 Search With Both Ship and Rescue Boat

In line 1060 see listing page E-4 and discussion above, the program made the final branch to determine which of the five paths for the remainder of the simulation to follow. The discussion above concerned one of the five paths, this discussion will take up another--search with both the rescue boat and the ship. This path begins at line 1200, page E-4, with a search first with the ship followed, at line 1230, by the concurrent search with the rescue boat. Line 1212 sets flag F(3) to two and the switch variable, Z, to three and line 1214 calls subroutine 7000, pages E-11 and E-12, to set up array T3. Then line 1216 divides the value of the navigational error stored in array T3 by two as prescribed by the Man Overboard Simulation Model. Next line 1218 calls subroutine 9400, page E-15, to perform the search. Lines 1212 through 1222 save the results of th search. Lines 1230 through 1238 test to see if the sea is too rough to launch the rescue boat and handle a rescue boat launch failure. Note that for a failure array element S8(3) gets incremented to keep a tally of rescue boat launch failures; flag F(4) receives the value four to identify the precise kind of failure in the flags; while the times are set to zero. Observe that a rescue boat launch failure, in this case, does not automatically mean a failure to rescue the MOB. For a rescue boat failure, control jumps to line 1250 while for a successful launch control jumps to line 1240, page E-4. In line 1240 the switch variable, Z, receives a value of four and the tally in array S8(4) gets incremented to accumulate the number of times the rescue boat got successfully launched. Lines 1242 through 1246 set up the array T3, adjust the navigational error for a search by both, and perform the search. In the path under discussion---search by both, regardless, of whether or not the rescue boat launch and/or search succeeded, control reaches line 1250. Lines 1250 and 1252 determine whether the ship found the MOB, jump to line 1256; only the rescue boat found the MOB, jump to line 1260; or neither found the

MOB, jump to line 3400. If neither found the MOB, lines 3400 through 3499, page E-7, set flag F(6) to four, test to see which vessel, if either, reached the MOB before he expired, and gather the statistics if either vessel reached the MOB before the end of the survival time. In any case line 5900, page E-8, receives the control soon after line 3400. Meanwhile for a successful search by ship, line 1256 determines if the rescue boat failed or the ship found the MOB first. If so control jumps to line 1270, otherwise control proceeds normally to line 1260. Control also reaches line 1260 from line 1254 when only the rescue boat finds the MOB. Line 1260 handles the cases in which only the rescue boat finds the MOB or the rescue boat finds the MOB first, or at precisely the same time as the ship, by setting flag F(5) to four and incrementing the running time by the time for the rescue boat to return to the MOB's position. Line 1262 then calls subroutine 6500 to gather the statistics. Lines 1264 through 1268 increment the running time by the time to find the MOB, gather the statistics, and jump to line 2350 to complete the simulation. If only the ship found the MOB or the ship found the MOB first, lines 1270 through 1276 perform the time incrementation and statistics gathering. Whenever the MOB has been found by the ship or, for searches by both ship and rescue boat, found by the ship first, control eventually gets to line 1400, page E-5. Lines 1400 through 1430 then compute and increment the running time by the time required for the ship to maneuver close to the MOB. Line 1440 tests to see if the MOB still lives when the ship gets nearby; if not, control jumps to line 4900, page E-8. Line 1450 transfers control to line 2100, page E-6.

#### 3.2.2.5 MOB Seen Going Overboard

In line 1030, page (30) above and listing page E-4, control jumped to line 1500 if the ship was restricted from turning. The discussion now considers line 1500 and following. Lines 1500 through 1520, page E-5, set the switch variable, Z to four and jump to line 1820, page E-5, to carry out a search with the rescue boat. The code beginning at line 1820 also applies to the other path which employs only the rescue boat for the search. On page (29) above, line 830 page E-3 of the program listing, the program tested whether or not the MOB was seen going overboard. If the MOB was seen going overboard control jumped to line 1600, which the discussion has just reached. Lines 1600 through 1695, page E-5, precisely parallels the code in lines 840 through 882 displayed on page E-3 and discussed on page (29) above. Lines 1700 through 1750, shown on page E-5, parallels lines 945 through 1050 which appear on page E-4. Note that the part of the program under discussion, lines 1600 and following, have no section comparable with lines 888 through 940 because the MOB was, in lines 1600 and following, seen going overboard. Note that line 1710 corresponds to line 990 except that line 1710 jumps to line 3600 instead of line 3200 if the MOB expires before the bridge receives notification. When the MOB was not seen, line 990, line 3200 and follow-sets flag F(6) to two; for the "seen" case, line 1710, line 3600, page E-7, and following sets flag F(6) to six. Line 1750 causes control to resume at line 2000 if the ship can turn.

### 3.2.2.6 Ship Restricted From Turning When MOB Seen

Lines 1800 through 1990 perform a search with the rescue boat when the ship cannot turn to search. The program in lines 1820 through 1990 also applies to a search by rescue boat only when the MOB was not seen going overboard. Control reaches line 1820, then, either by normal progression from line 1810 or by jump from line 1520. Note that line 1510, page E-5, sets the switch variable, Z, to four while line 1810 sets Z to two. The switch variable, Z, determines the values used for the search, see subroutine 7000 on pages E-11 and E-12. Line 1820 sets flag F(2) to two to indicate search by boat. Lines 1830 through 1860 test the sea state against the sea worthiness of the rescue boat and handle the condition of rescue boat launch failure by a jump to line 3700, page E-7. Lines 3700 through 3799 set flag F(6) to seven, increment the rescue boat launch failure tally, S8(3), and jump to the common exit line, 5900, on page E-8. For a successful rescue boat launch, lines 1870 through 1895 set up the array T3 by a call to subroutine 7000, perform the search, and increment the running time to the point at which the rescue boat reaches the MOB's position. Lines 1900 through 1910 set flag F(3) to two, increment the "rescue boat deployed" flag, S8(4), and test for a successful search. If the search failed control jumps to line 3800, page E-8, where flag F(6) gets set to eight. Continuing the discussion of a rescue boat search failure, line 3830 tests to see if the MOB expired before the rescue boat reached his position. If the MOB had expired by the time the rescue boat returns to the MOB, control goes to the common exit line. Otherwise line 3840 gathers the statistics first. For a successful search lines 1930 through 1990, page E-6, complete the statistics gathering for both the point in the search at which the rescue boat reaches the MOB's position and the finding of the MOB. Line 1990 causes a jump to line 2350, page E-6, to complete the rescue.

### 3.2.2.7 Ship Not Restricted From Turning When MOB Seen

At line 1750, page E-5, the program decided whether or not the ship was restricted from turning. If the ship could not turn, the program discussion above followed; the discussion below applies if the ship could turn and search. Lines 2000 through 2085, page E-6, perform this search and gather the statistics. Except for setting flags to different values, F(2) and F(3) set to unity in this case, the switch variable, Z, getting a unity value instead of two, and a different jump point for a search failure, 3900 instead of 3800, the code for a search with ship exactly equals the code for search with rescue boat. Note that at 3920, page E-7, the flag F(6) gets set to nine. However, the code in lines 2100 and following has no parallel in the case of a search with rescue boat. Line 2100 may be reached either by normal progression from line 2085 or by a jump from line 1450, page E-5. This code handles the case of a successful search by ship.

### 3.2.2.8 MOB Found By Ship

When the ship successfully finds the MOB the captain decides whether to use the rescue boat to bring the MOB back aboard or recover the MOB directly. Currently the Man Overboard Simulation Model, reference 1, performs this decision by providing a probability which depends upon the ship type. The probability, P10, appears as ship characteristic number eleven, T1(11), and P2(10). If the user desires different values, the Man Overboard Operator Program requires modification. The DATA statements in lines 9910 through 9980, page C-4, contain probability P10 as the eleventh numerical value (there are twelve numerical values in each DATA statement). In the program under discussion, the Simulation Program, line 2110 uses probability P10 with the random number generator to simulate the captain's decision. For recovery with the rescue boat, control jumps to line 2200. Otherwise control advances normally to line 2120. If the program decides to recover the MOB directly, line 2130 determines whether or not the ship's efforts succeeded. A failure here results in a jump to line 4100, page E-8, which sets flag F(6) to ten and jumps to the common EXIT point at line 5900. On the other hand, a success results in a jump to line 5000--the first of three success exit points.

### 3.2.2.9 Recover of MOB With Rescue Boat

For a recovery by rescue boat, the program goes to line 2200 where a more complicated set of code follows. First the program tests flag F(3). This flag tells whether or not the rescue boat has already been launched. The test causes a jump to line 2300 if the rescue boat has not been launched. Control continues to line 2230 if the rescue boat got successfully launched during a search with both rescue boat and ship. Lines 2230 through 2280 determine and increment the running time by the time for the rescue boat to return to the MOB. Note that this provides two different formulas for the time for the rescue boat to reach the MOB depending upon whether or not the ship found the MOB on the first pass. Lines 2285 and 2290 cause a jump to either line 4200, if the MOB expired before the rescue boat returned or line 2350 if the MOB was reached in time. In the event the rescue boat had not been launched successfully, line 2220 caused a jump to line 2300. Lines 2300 through 2310 determine whether or not a successful launch can occur and jump to line 4300 for a failure. Note that if the search had been attempted by both, the rescue boat launch had failed, and the ship successfully found the MOB, line 2310 will now record the launch failure. Lines 2330 and 2335 increment the running time by the amount of time required to launch the rescue boat and the tally of successful rescue boat launches, S8(4), by one. Line 2340 detemines whether or not the MOB remains alive after the rescue boat launch completes. If not, control jumps to line 4400, page E-8.

3.2.2.10 Maneuver Rescue Boat NEAR MOB

Next lines 2350 through 2390 compute the time required to maneuver the rescue boat near the MOB, increment the running time, and determine whether or not the MOB still lives. If the MOB did not survive until the rescue boat maneuvered nearby, control jumps to line 4500 on page E-8. Note that control can reach line 2350 either by normal progression from line 2340 or by a jump from lines 1266, 1999, or 2290 on pages E-5 and E-6. At this point the rescue boat has pulled along side the MOB.

3.2.2.11 Bring MOB Aboard Rescue Boat

Lines 2400 through 2450 compute the time required to bring the MOB aboard the rescue boat, increment the running time, and test the running time against the survival time. If the MOB does not survive the time required to come aboard the rescue boat, control jumps to line 4600. Lines 2460 and 2490 gather the statistics at the time rescue boat successfully pulls the living MOB aboard. At this point the survival time clock stops running. Therefore, the time for the rescue boat to return to the ship does not affect the simulated time.

3.2.2.12 Bring MOB Aboard Ship

The MOB is not considered "rescued" until back aboard the ship. The MOB can get aboard the ship either by a successful rescue boat recovery or by other means. The Man Overboard Simulation Model, reference 1, has provision for recovery by either means as determined by probability P13. Probability P13 has a value of zero so the only way the MOB can get back aboard the ship is by a successful rescue boat recovery. The Man Overboard Simulation Program implements the recovery in lines 2500 through 2570. Lines 2520, 2540, and 2560 use probabilities P13, P14, and P15 restively along with the random number generator to accomplish these final switches. Successfully getting the rescue boat, with crew and MOB, aboard causes a jump to line 5200, page E-7, at line 2560. At line 2570 failure results in a jump to line 4800. Pages E-7 and E-8 contain the part of the program that handles the successes and failures. Jumps to line 5100 and 5200 cause flag F(6) to get set to 21 or 22 respectively. For either success lines 5230 and 5260 gather the statistics. All successes ultimately execute line 5270 which accumulates the total number of MOB's saved regardless of whether or not the rescue boat partipated in the rescue. Note that flag F(6) may take on any one of 22 different values with values one through eighteen different types of failures, value twenty indicating recovery by ship without using the rescue boat to recover the MOB, and values 21 and 22 indicating successful recovery using the rescue boat. All the points to which the program jumps to indicate a success or a failure appear on pages E-7 and E-8 in lines 3100 through 5270.

### 3.2.3 Output Section

Eventually all simulations transfer control to line 5900, page E-8. Lines 5900 through 5940 produce a print-out for each simulation, if the user selected debug print #2. Line 5999 terminates the FOR-NEXT loop that began with line 480 on page E-2. Unless the user requests one or more of the debug print-outs, no print-out occurs until after control passes line 5999 to line 6000. The Man Overboard Simulation Program provides two different print-outs summarizing the results of the simulations. Lines 6000 through 6020 produce a one page report on the simulations containing a summary of both the input data and the results of the simulations. The user requests this one page report by requesting a "REGULAR PRINT". The PRINT statements appear as subroutine 6100 which appears as lines 6100 through 6499 on pages E-9 through E-11. If the user requested a "COMPRESSED PRINT", lines 6026 through 6066 produce a summary of the results of the simulation (without a print-out of the input data) in five lines. Finally, lines 6070 through 6090 write the results to the disk file opened in line 156 on page E-1. Note that lines 160 through 262 write the input data to the output file so the output file contains the input data followed by the results of the simulations. The last line of the main program, line 6099 on page E-8, transfers control back to line 140 on page E-1 to read the next group of input data from the input disk file.

### 3.2.4 Subroutines

The main program calls a number of subroutines which require discussion. The subroutines begin with subroutine 6100 on page E-9. Subroutine 6100 produces the "REGULAR PRINT". Subroutine 6500 on page E-11 accumulates the statistics for ten different events--five each for the "seen" and "not seen" cases. Subroutine 7000, pages E-11 through E-13, sets up an array, T3, containing all the parameters needed to perform a search. Subroutine 7800 on page E-13 prints the heading for the "REGULAR PRINT" while subroutine

7900 produces the "COMPRESSED PRINT" heading. The subroutine that begins with line 8000 selects a variant from a normal distribution having a specified mean, minimum, maximum, and standard deviation. A call to subroutine 8100 selects a variant from a normal distribution having the mean specified with a minimum of zero, a maximum of twice the mean, and a standard deviation of one-half the mean. This results in a normal distribution having the specified mean with a range of two sigma ($2\sigma$) on each side of the mean. The following subroutine, subroutine 8200 provides the program with a Rayleigh distribution, reference 3, while subroutine 8300 makes a uniform distribution available. Subroutines 8400 and 8500 compute the survival time due to and floatation respectively. Subroutines 8700 and 8800 produce debug print-outs #1 and #2 respectively. Subroutine 9000 gives values to a number of variables which remain constant throughout the simulation runs. The little short subroutine beginning with line 9200 homes and erases the video display. The very core of the entire simulation consists of the events from the time the bridge receives notification of a man overboard condition until the MOB is spotted in the water. The core of the simulation appears as subroutine 9400 which receives its input parameters in array T3 and puts its results in flag F(4), and variables T2, T4, and K. Finally the distance of visibility from both the ship and the rescue boat gets computed in subroutine 9500.

### 3.2.4.1 Subroutine 6100: Regular Print

Subroutine 6100 uses a number of un-formatted and formatted (PRINT USING) statements to produce a one-page report containing both the input data and the results of a specified number of simulations. The user invokes the print-out by requesting a "REGULAR PRINT". Line 6125 defines the format for the two lines of the input data part of the report that use formatted PRINT statements, The two lines of input data that use formatted PRINT appear in lines 6154 and 6157 on page E-9. Line 6130 on page E-9 sets up a number of tab stops for use through the remainder of the subroutine. Lines 6140 and 6184 print the input data using un-formatted PRINT statements for all lines except the two noted above. Line 6186 begins the print-out of the results of the simulations by declaring how many simulations constituted the simution run. Line 6188 simply produces two blank lines. Lines 6190 and 6192 compute the overall fraction of MOB's rescued by the rescue boat and establish the format with which to print the value. Lines 6194 through 6220 print the lines containing the overall fraction of MOB's rescued by the rescue boat. Since this is a key result of the simulation, the value receives a distinctive treatment by providing a surrounding box of asterices (*). Next, two lines appear, 6222 and 6224, to print the number of times the rescue boat was deployed and the launch failed due to sea state followed by two blank lines. Lines 6252 through 6254 print the number of times the MOB was seen and not seen going overboard. Note that line 6254 print a colon (:) two columns ahead of the title "MOB NOT SEEN". A colon appears in this same column on each line in the remainder of the print-out to separate the "not seen" results from the "seen" results. Line 6260 produces two lines that are blank except for the colon discussed above. Lines 6262 and 6264 produce the headings for the columns that constitute the remainder of the report page. Note that the two halves--"seen" and "not-seen"--of the report have identical headings except for colon separating the two halves. Lines 6266 and 6268 print the heading line "MOB ALIVE WHEN:" preceeded and followed by a line blank except for the colon. Lines 6270 through 6292 print five lines of information with each line of information followed by a line blank except for the colon separating the "seen" and "not seen" information. The text identifying the numerical information on the line gets printed by the PRINT statements in lines 6270, 6274, 6280, 6286, and 6290. Subroutine 6400, lines 6400 through 6499, prints the numerical information using the PRINT USING statements in lines 6470, 6471, and 6474. Note that the variable J8 receives a value before each subroutine call to determine what information the subroutine will print. The subroutine at 6400 prints the "seen" information followed by a colon and then the "not seen" information. The subroutine follows each line of information by a line blank except for the colon.

### 3.2.4.2 Subroutine 6500: Accumulate Statistics

The subroutine on page E-11 that begins with line 6500 accumulates the statistics for the simulations in a run. Variable J9, an input to the subroutine, determines which set of statistics to update. The array S9 accumulates a count of the number of times an event occurs. Array T9 sums the time at which the event occurs. After the run completes, the total time gets converted into the mean time by dividing by the count in S8. Arrays N9 and M9 keep the minimum and maximum time respectively. Note that values of J9 from one through five apply to the cases in which the MOB was seen going overboard while J9 equal to six through ten accumulate values when the MOB was not seen. Line 6530 tests J9 to see if it is greater than five and jumps to line 6560 if it is. Line 6540 tests the flag, F(1), to determine whether the current case is a "seen" and if so jumps to line 6560. Line 6550 increments J9 by five. Thus if subroutine 6500 gets called with "not seen" information with J9 less than five (which applies to "seen" cases), line 6550 makes J9 five greater so the remainder of the subroutine will update the proper elements in the arrays. This arrangement allowed parts of the program to share code for "seen" and "not seen" cases.

### 3.2.4.3 Subroutine 7000: Initialize Search

Subroutine 7000 begins on page E-11 and concludes on page E-13. The subroutine sets up array T3 to provide the input parameters to the search routine beginning at line 9400. Table I below shows the values computed for each element of the array for each of the four possible types of searches which subroutine 9400 will perform. The subroutine uses variable Z as a switch variable to determine which set of equations to use. Note that the four types of search correspond to four of the five paths the simulation may follow. The fifth path, search by both ship and rescue boat when the MOB was not seen going overboard, consists of a special combination of the "not seen" cases from Table I below. The special combination consists of a search by ship with the navigation error set to one-half the value normally computed followed by a search using the rescue boat with the diminished navigational error. The subroutine 7000 calls subroutines 7200, 7300, or 7400 depending upon the value of the switch variable, Z, to compute the values of T3(1) and T3(2). The subroutine continues at line 7100 for the "seen" cases and at line 7500 for the "not seen" cases.



49

TABLE I: Variables in Array T3

| | Man seen going overboard | | Man not seen going overboard | |
|---|---|---|---|---|
| | Search with Ship | Search with Boat | Search with Ship | Search with Boat |
| Switch Variable | 1 | 2 | 3 | 4 |
| Subscript for T3 | | | | |
| 1 | $T_T$ | $\mathrm{Max}(T_{SL}, T_{RPT}+T_p)$ | $T_{TW}$ | $\mathrm{Max}(T_{SL}, T_{RPT}+T_p)$ |
| 2 | $\frac{1}{2}T_{SL}+T_N$ | $T_L+\left[\frac{V_S}{V_B}\right](\frac{1}{2}T_{SL}+T_N)$ | $T_M+T_N$ | $T_L+(T_M+T_N+\frac{1}{2}T_{SL})\frac{V_S}{V_B}$ |
| 3 | $T_S-T$ | $T_S-T$ | $T_S-T$ | $T_S-T$ |
| 4 | Navigational Error | Navigational Error | Navigational Error | Navigational Error |
| 5 | 0 | 0 | $d_{V_S}$ | $d_{V_B}$ |
| 6 | 0.5 | 0.5 | $T_{LS}$ | $T_{LS}$ |
| 7 | 0 | 0 | $T_T$ | $T_T$ |
| 8 | 0.5 RND(1) | 0.5 RND(1) | $T_M$ | $T_M$ |
| 9 | T3(3)-T3(1)-T3(2) | $\frac{\mathrm{T3(3)-T3(1)-T3(1)}}{\mathrm{V(4)}}$ | $\frac{\mathrm{T3(3)-T3(1)-T3(1)}}{T_{LS}+T_T}$ | $\frac{\mathrm{T3(3)-T3(1)-T3(2)}}{\mathrm{V(4)}(T_{LS}+T_T)}$ |
| 10 | $P_{12}=0.9$ | $P_{12}=0.9$ | $P_8=0.5$ | $P_8=0.5$ |

#### 3.2.4.4 Subroutine 7600: Debug Print #3

Subroutine 7600 performs debug print #3. The subroutine, appearing on page E-13, prints a heading, some of the ship characteristics, and the rescue boat characteristics. All these values are simply part of the input data; all remain constant for all simulations in a run.

#### 3.2.4.5 Subroutine 7800: Regular Print Heading

On page E-13 subroutine 7800 appears which prints the heading for the regular print-out. The subroutine consists entirely of simple PRINT statements.

#### 3.2.4.6 Subroutines 8000, 8100, 8200, and 8300: Distributions

On page E-13 and E-14 the subroutines for the various distributions appear. Each of these subroutines follow precisely the similar subroutines in the Abandon Ship Simulation Program. For each of the distribution subroutines, the input variables appear in array R while the selected variant appears in variable R3. For the normal distribution subroutine which begins in line 8000, the mean inputs in R(1), the minimum comes from R(2), the maximum appears in R(3), and R(4) contains the standard deviation. The special distribution which begins at line 8100 receives a mean in R(1) and computes values for R(2), R(3), and R(4) then jumps to subroutine 8000 to select a variant from the normal distribution and return to the calling program. Subroutine 8100 sets R(2) (minimum) to zero, R(3) (the maximum) to twice the mean, in R(1) and R(4) (the standard deviation) equal to one-half the mean. Thus subroutine 8100 produces a normal distribution having a two sigma ($2\sigma$) range on each side of the mean. Subroutine 8200 selects a variant from a Raleigh distribution given the mode in R(1), see reference 3. The last of the distributions begins at line 8300 on page E-14 and concludes with line 8399. The uniform distribution, subroutine 8300, requires the minimum in R(2) and the maximum in R(3). Note these are the same element numbers as for the corresponding values in the normal distribution.

3.2.4.7 Subroutines 8400 and 8500: Survival Time

Subroutines 8400 and 8500 on page E-14 compute the components of the survival time. Subroutine 8400 computes the survival time due to hypothermia by setting up the inputs for a normal distribution then jumping to line 8000. The minimum and maximum follow from a curve fit to a published curve (actually a band having an upper and lower bound), see reference 4, while the mean and standard deviation appear as the mean, between minimum and maximum, and the assumption that the upper and lower bound constitute two sigma ($2\sigma$) deviations from the mean. The subroutine starting on line 8500 establishes the survival time due to flotation. The subroutine requires the probability the MOB was wearing a PFD in variable P, the mode of the survival time without flotation in array element T5(3), and the mode of the survival time with flotation in array element T5(4). Line 8510 uses the random number generator to determine whether or not the MOB has flotation. Note that lines 8530 and 8570 set flag F(2) to one for flotation and two for no flotation. Regardless of whether or not the MOB has flotation, the subroutine jumps to the Rayleigh distribution, line 8200, to select the survival time and return to the main program.

3.2.4.8 Subroutines 8700 and 8800: Debug Prints #1 and #2

Subroutine 8700 produces the debug print-out when the user requests "DEBUG PRINT #1". The routine consists mostly of PRINT statements. The print-out prints the variables that remain constant during a given simulation with appropriate headings to identify the variables. Since the call to this subroutine lies within the loop that determines how many simulations will be done, this subroutine can produce a large amount of output if enabled. Line 8745 converts the variable S into a string and truncates it to fifteen characters. Thus lines 8745 and 8750 could have employed a PRINT USING statement. This subroutine gets called near the beginning of each simulation. Subroutine 8800, on the other hand, gets called at the end of each simulation when enabled by the user requesting "DEBUG PRINT #2". Subroutine 8800 uses the format lines 8820 and 8830 in the PRINT USING statements in lines 8845 and 8830. Lines 8840 through 8860 print the six flags, F(1) through F(6). Line 8878 prints the running time at the end of the simulation, T, the survival time T0, the last value returned by a distribution, R3, and the wave height, V(2). The print-out prints one line for each simulation; therefore for a run of 1000 simulations (typical number in a run) the print-out would produce 1000 lines of output if the user selected "DEBUG PRINT #2".

3.2.4.9 Subroutine 9000: Initialize Constants

Subroutine 9000 begins on page E-14 and concludes on page E-15. This subroutine provides values for a number of variables that remain constant throughout all the simulations in a run. The subroutine consists almost entirely of assignment statements. Note that for the most part the variables appear in alphabetical numerical order. Variable B9 gets set to a value near the largest the machine can represent. Variables C0 through C4 receive values equal to their terminating digit. The variables in array D0 no longer have any use in the program. Variable L9 has a value near the smallest positive value the machine can represent. Variables P3 through P6 constitute a set of variables used to compute probability P15. Variables S1 through S6 constitute a set of constants for the survival time due to hypothermia calculation. Lines 9090 through 9098 set up variable B$ to a string of blanks, but B$ no longer fills any requirement. Line 9099 returns control to the main program.

3.2.4.10 Subroutine 9200: Home and Erase Video Display

Subroutine 9200, appearing on page E-15 simply clears and erases the video display. This simple function was put into a subroutine so that conversion to different machines of the program would be simplified. Each type of machine requires a different code to accomplish this function.

3.2.4.11 Subroutine 9400: Return and Search

Subroutine 9400 constitutes the core of the simulation. The subroutine appears on page E-15. The subroutine handles the simulation from the time the bridge gets notified of the MOB until the ship finds or fails to find the MOB. The subroutine first computes the time for the ship or rescue boat to return to the MOB's position, line 9430. The test in line 9432 causes a jump to line 9465 if the MOB did not survive until the ship or rescue boat returned. Line 9434 tests to see if the MOB will be in visible range when the ship returns; if so control goes to line 9480. Line 9436 tests to see if the MOB will expire before the end of one search pass; if so, control jumps to line 9475. Lines 9440 through 9452 perform the multi-pass search with a successful find causing a jump to line 9490 and a failure going either to line 9470 or 9460. Lines 9460 through 9490 receive the various branches as a result of success or failure. Each of these lines sets flag F(4) to a different value. Values less than eight indicate a failure while values 8, 9, and 10 indicate a success. The specific value tells how the success or failure occured.

3.2.4.12 Subroutine 9500: Distance of Visibility

Subroutine 9500, on page E-15, computes the visibility from the ship and from the rescue boat. The equations use the height of the lookout's eye, T1(4) for the ship D1(5) for the rescue boat, the environmental visibility, T5(15), and a maximum visibi-

# APPENDIX A: Abandon Ship Simulation Program Listing

```
5 DIM X1(11),T9$(11)64,D1(16),M5$(28)64,T9(8,11),N5(21),V1(12),P1(15),T5(10)
6 DIM C0$(10)64,C0(9,10),Y1(11,3),M6(28),M4(28),R1(12),A$64,A1$64,M7$64,P8$64
7 DIM V$64,Z0$64
1000 REM ABANDON SHIP SIMULATION PROGRAM
1002 GOSUB 9000
1004 A1$="ABANDON SHIP SIMULATION"
1006 GOSUB 9000:PRINT A1$
1008 INPUT "SERIAL NUMBER",S9
1010 M=INT(S9)
1012 PRINT "INITIALIZING"
1014 GOSUB 1504
1016 FOR I=C1 TO M:Z1=RND(C1):NEXT I
1018 GOSUB 1046:REM SHIP CH.
1020 GOSUB 1102:REM RESCUE BOAT CH.
1022 GOSUB 1134:REM CASUALTY TYPE.
1024 GOSUB 1200:REM PRINT INPUT DATA .
1025 Z2=0
1026 W=Z2
1028 FOR K1=Z2 TO X1(Z2+1)
1030 E9=Z2:X1(C2+1)=Z2
1032 GOSUB 1674
1034 IF E9<Z2 THEN 1044
1036 GOSUB 1860
1038 IF E9=Z2 THEN 1039:GOTO 1040
1039 GOSUB 2044
1040 NEXT K1
1042 GOSUB 1266
1044 GOTO 1000
```

```
1504 REM
1514 RESTORE
1516 K=1:PRINT
1518 PRINT HEX(0D)
1520 READ  P1(14),P1(15)
1530 REM ?
1532 FOR I=1TO 10:READ T5(I):NEXT I
1534 REM CASUALTY TYPES
1536 READ  M3
1540 FOR I=1TO M3+1
1542 READ C0$(I)
1544 FOR J=1TO 10
1546 READ C0(I,J)
1548 NEXT J:NEXT I
1550 REM SHIP TYPES
1552 READ M9
1556 FOR I=1TO M9+1
1558 READ T9$(I)
1560 FOR J=1TO 11
1562 REM ?
1564 READ T9(I,J)
1566 NEXT J:NEXT I
1568 REM RESCUE BOAT
1570 READ M8
1572 M1=M8+19
1576 FOR I=1TO M1+1
1578 REM ?
1580 READ M5$(I),M6(I),M4(I)
1582 NEXT I
1584 B5=999
1586 C1=1:C2=2:C3=3:C4=4
1588 C5=.5
1590 C6=6:C7=7:C8=8:C9=9
1592 Z0=60
1594 Z0$="CHARACTERISTICS."
1596 Z5=.25
1598 L5=9.5E-39
1600 P3=100
1602  P0=#P1
1604 P7=C2*P0
1606 S1=.0024:S2=8.1E-6
1608 S3=.1:S4=5.9E-5
1610 S5=.101:S6=2.25
1614 Z2=0
1616 Z1=RND(-C1)
1618 RETURN
```

```
1372 REM P1
1374 DATA 0.02,0.5
1378 DATA 15.0,0,5,0,30,10,0,2,2,0.2:REM T5
1380 REM C0
1382 DATA 8:REM MAX. INDEX
1384 DATA "FIRE",0,0,0.05,0,0.05,0.1,0,0,30,1
1386 DATA "COLLISION & FIRE",1,0,0.02,0.5,0.05,0.1,20,0,30,2
1388 DATA "COLLISION & SINKING",0,0,0.02,0.5,0,0,20,20,30,2
1390 DATA "EXPLOSION & FIRE",0,1,0.1,1,0.05,0.1,0,0,5,3
1392 DATA "EXPLOSION & SINKING",0,0,0.1,1,0.05,0.1,0,20,20,3
1394 DATA "STRUCTURAL FAILURE",0,0,0,0,0.05,0,0,20,15,4
1396 DATA "GROUNDING",0,0,0,0,0,0,0,20,20,5
1398 DATA "CAPSIZING",0,0,0.1,2,0,0,0,30,5,6
1400 DATA "FOUNDERING",0,0,0,0,0,0,0,20,20,5
1402 REM T9
1404 DATA 7:REM MAX. INDEX
1406 DATA "TANKER",845,40,4,33,1,25,6.5,3,38,70,0.2
1408 DATA "CONTAINER",710,40,2,40,1,25,5,3,38,70,0.3
1410 DATA "LNG",900,50,4,33,2,25,5,3,38,70,0.25
1412 DATA "FISHBOAT",75,5,0,0,1,10,5,3,38,70,0.75
1414 DATA "TUG",150,10,0,0,1,10,5,3,38,70,0.8
1416 DATA "BARGE CARRIER (LASH)",740,45,2,50,1,25,5,3,38,70,0.3
1418 DATA "GREAT LAKES BULKER",708,33,2,40,2,10,4,2,42,70,0.4
1420 DATA "FERRY",150,200,0,33,8,25,5,3,38,70,0.8
1422 REM RESCUE BOAT
1424 DATA 7:REM MAX. INDEX
1426 DATA "NO. OF RESCUE BOATS (1 OR 2)",1,2
1428 DATA "BOAT CAP.(EXC. CREW)",1,50
1430 DATA "NO. IN BOAT CREW",1,10
1432 DATA "MEAN TIME TO LAUNCH (MIN.)",0,60
1434 DATA "MEAN TIME TO PICKUP (MIN.)",0,60
1436 DATA "STD. MANEUVER TIME (MIN.)",0,60
1438 DATA "MAX. WAVE HEIGHT (FT.)",0,101
1440 DATA "MAX. HEEL ANGLE (DEG.)",0,90
1444 REM CHARACTERISTICS FOR OVERRIDE
1445 DATA "(1=YES; 0=NO)",0,1
1446 DATA "SHIP LENGTH",20,5000
1448 DATA "POB SHIP",2,5000
1450 DATA "NO. OF LIFEBOATS",0,200
1452 DATA "CAPACITY OF EACH L.B.",0,100
1454 DATA "NO. OF LIFERAFTS",0,200
1456 DATA "CAPACITY OF EACH L.R.",0,100
1458 DATA "MODE OF SIGN. WAVE HGTS.--MID-WINTER",0,50
1460 DATA "MODE OF SIGN. WAVE HGTS.--MID-SUMMER",0,50
1462 DATA "MEAN WATER TEMP.--MID-WINTER",20,100
1464 DATA "MEAN WATER TEMP.--MID-SUMMER",20,100
1466 DATA "PROB. RESCUE VESSEL NEARBY",0,1
1468 DATA "MEAN FRACTION OF POB KILLED",0,1
1470 DATA "MEAN NO. OF POB KNOCKED OVERBOARD",0,1000
1472 DATA "MEAN FRACTION OF POB ISOLATED",0,1
1474 DATA "MEAN FRACTION OF LIFEBOATS DESTROYED",0,1
1476 DATA "MEAN LENGTH OF DAMAGE",0,200
1478 DATA "MEAN HEEL ANGLE",0,180
1480 DATA "MEAN TIME FOR CASUALTY TO DEVELOP",0,480
```

```
1046 REM
1048 GOSUB 9000
1050 PRINT "SHIP";" ";Z0$
1052 FOR I=Z2 TO M9
1054 PRINT I+C1;T9$(I+1)
1056 NEXT I
1058 INPUT "NUMBER",N9
1060 N9=INT(N9-C1)
1062 IF N9<Z2 THEN 1058
1063 IF N9>M9THEN 1058
1064 PRINT "WANT STANDARD SHIP";
1066 I5=Z2:J1=M8+C1:GOSUB 1620
1068 IF D1(Z2+1)=C1 THEN 1086
1070 IF D1(Z2+1)<>Z2 THEN 1064
1072 I5=Z2:K1=Z2
1074 FOR J1=M8+C2 TO M8+12
1076 PRINT :PRINT "     (STD. ";M5$(J1+1);" =";T9(N9+1,K1+1);")";
1078 GOSUB 1620
1080 T9(N9+1,K1+1)=D1(K1+1)
1082 K1=K1+C1
1084 NEXT J1
1086 N5(Z2+1)=T9(N9+1,C1+1)
1088 N5(14)=T9(N9+1,C2+1)
1090 N5(15)=T9(N9+1,C4+1)
1092 N5(16)=T9(N9+1,C3+1)
1094 N5(17)=T9(N9+1,6)
1096 V1(C2+1)=C7*(T9(N9+1,C3+1)↑(C1/C3))
1098 P1(C4+1)=T9(N9+1,11)
1100 RETURN


1102 REM
1104 I5=Z2
1106 GOSUB 9000
1108 PRINT "RESCUE BOAT";" ";Z0$
1110 FOR J1=Z2 TO M8
1112 GOSUB 1620
1114 NEXT J1
1116 REM RESCUE BOAT LENGTH.
1118 V1(C3+1)=C7*((D1(C1+1)+D1(C2+1))↑(C1/C3))
1120 N5(12)=D1(Z2+1)
1122 REM PREPARE FOR P8.
1124 V1(C4+1)=C1/T9(N9+1,Z2+1)
1126 V1(6)=C5*(T9(N9+1,C2+1)*V1(C2+1) + V1(C3+1)*N5(12))
1128 N5(C4+1)=D1(C2+1)
1130 T5(C8+1)=D1(C5+1)
1132 RETURN


1620 REM
1622 PRINT :PRINT "    ";M5$(J1+1);
1624 INPUT D1(I5+1)
1626 IF D1(I5+1)<M6(J1+1) THEN 1622:IF D1(I5+1)>M4(J1+1)THEN 1622
1628 I5=I5+C1
1630 RETURN
```

```
1134 REM
1136 GOSUB 9000
1138 PRINT "TYPE OF CASUALTY"
1140 FOR I=Z2 TO M3
1142 PRINT I+C1;C0$(I+1)
1144 NEXT I
1146 INPUT "NUMBER",T0
1148 T0=INT(T0-C1)
1150 IF T0<Z2 THEN 1146
1151 IF T0>M3THEN 1146
1152 REM
1154 PRINT "WANT STANDARD CASUALTY";
1156 L1=I5:K1=C2
1158 J1=M8+C1:GOSUB 1620:I5=L1
1160 IF D1(I5+1)=C1 THEN 1172
1162 IF D1(I5+1)<>Z2 THEN 1152
1164 FOR J1=M8+13 TO M1
1166 PRINT :PRINT "     (STD. ";M5$(J1+1);" =";C0(T0+1,K1+1);")";
1168 GOSUB 1620:C0(T0+1,K1+1)=D1(I5-C1+1)
1170 K1=K1+C1:NEXT J1
1172 P1(C3+1)=C0(T0+1,Z2+1)
1174 P1(6)=C0(T0+1,C1+1)
1175 IF T0>Z2THEN 1176:GOTO 1178
1176 IF T0<C3THEN 1177:GOTO 1178
1177 P1(C4+1)=.98
1178 N3=C0(T0+1,C9+1)
1180 INPUT "NUMBER OF SIMULATIONS",X1(Z2+1):REM -------------------------
1182 X1(Z2+1)=INT(X1(Z2+1)-C1)
1184 IF X1(Z2+1)<Z2 THEN 1180
1186 FOR I=Z2 TO 10
1188 Y1(I+1,Z2+1)=Z2
1190 Y1(I+1,C1+1)=B5
1192 Y1(I+1,C2+1)=-B5
1194 NEXT I
1196 Y1(C1+1,Z2+1)=-C1
1198 RETURN
```

```
1674 REM SECTION A *************
1676 L2=Z2:L=Z2
1678 ON N3 GOTO 1680,1690,1714,1734,1806,1750
1680 REM CASUALTY = FIRE
1682 N5(C3+1)=Z2
1684 N5(C2+1)=Z2
1686 N5(C1+1)=Z2
1688 GOTO 1774
1690 REM CASUALTY = COLLISION
1692 R1(Z2+1)=C0(T0+1,C3+1):GOSUB 2258
1694 N5(C1+1)=INT(R6+C5)
1696 N5(C2+1)=Z2
1698 R1(Z2+1)=C0(T0+1,C2+1):GOSUB 2258
1700 N5(C3+1)=INT(N5(Z2+1)*R6+C5)
1702 R1(Z2+1)=C0(T0+1,C6+1):GOSUB 2258
1704 P1(C8+1)=V1(C4+1)*(V1(6)+R6)
1706 IF RND(C1)<=P1(C8+1)THEN 1708:GOTO 1710
1708 L2=C1
1710 IF RND(C1)<=P1(C3+1)THEN 1774
1712 GOTO 1768
1714 REM CASUALTY = EXPLOSION
1716 R1(Z2+1)=C0(T0+1,C3+1):GOSUB 2258
1718 N5(C1+1)=INT(R6+C5)
1720 R1(Z2+1)=C0(T0+1,6):GOSUB 2258
1722 L2=INT(R6*(N5(12)+N5(14))+C5)
1724 L=R6
1726 R1(Z2+1)=C0(T0+1,C4+1):GOSUB 2258
1728 N5(C2+1)=INT(N5(Z2+1)*R6+C5)
1729 R1(Z2+1)=C0(T0+1,C2+1):GOSUB 2258
1730 N5(C3+1)=INT(N5(Z2+1)*R6+C5)
1731 IF RND(C1)<=P1(6)THEN 1774
1732 GOTO 1768
1734 REM CASUALTY = STRUCTURAL FAILURE
1736 R1(Z2+1)=C0(T0+1,C4+1):GOSUB 2258
1738 N5(C2+1)=INT(N5(Z2+1)*R6+C5)
1740 N5(C3+1)=Z2
1742 GOTO 1764
1744 REM CASUALTY = GROUNDING
1746 N5(C2+1)=Z2:N5(C3+1)=Z2
1748 GOTO 1764
```

```
1750 REM CASUALTY = CAPSIZING
1752 R1(Z2+1)=C0(T0+1,4):GOSUB 2258
1754 N5(C1+1)=INT(R6+C5)
1756 N5(C2+1)=Z2
1758 R1(Z2+1)=C0(T0+1,C2+1):GOSUB 2258
1760 N5(C3+1)=INT(N5(Z2+1)*R6+C5)
1762 GOTO 1768
1764 N5(C1+1)=Z2
1766 REM FLOODING  ****************
1768 R1(Z2+1)=C0(T0+1,C7+1):GOSUB 2258
1770 T7=R6
1772 GOTO 1788
1774 REM FIRE **************
1776 R1(Z2+1)=C0(Z2+1,6):GOSUB 2258
1778 L=L+R6
1780 L2=INT(L2+R6*(N5(12)+N5(14)-L2)+C5)
1782 R1(Z2+1)=C0(Z2+1,C2+1):GOSUB 2258
1784 N5(C3+1)=N5(C3+1)+INT(N5(Z2+1)*R6+C5)
1785 R1(Z2+1)=C0(Z2+1,C4+1):GOSUB 2258:N5(C2+1)=N5(C2+1)+INT(N5(Z2+1)*R6+C5)
1786 T7=Z2
1788 L=C1-L
1790 N5(11)=N5(C1+1)+N5(C2+1)
1792 S=Z2
1794 X1(C1+1)=N5(11)
1796 J3=Z2:Z1=X1(C1+1):GOSUB 1662
1798 PRINT N5(11);
1800 GOTO 1818
1802 E9=15
1804 RETURN
1806 REM GROUNDING OR FOUNDERING **********
1808 PRINT C0$(C6+1);" OR ";C0$(C8+1)
1810 SELECT PRINT 215(160):PRINT :PRINT :PRINT TAB(C9);"NO SIMULATIONS POSSIBLE"
1812 PRINT "RESULTS IN NOBODY IN THE WATER."
1814 SELECT PRINT 005(80):E9=-15
1816 RETURN


1818 REM SEASON.
1820 R1(C1+1)=Z2
1822 R1(C2+1)=C1
1824 GOSUB 2288
1826 REM WAVE HEIGHT
1828 R1(Z2+1)=T9(N9+1,C6+1)*U9+T9(N9+1,C7+1)*(C1-U9)
1830 GOSUB 2294
1832 V1(C1+1)=R9
1834 REM WATER TEMPERATURE
1836 R1(Z2+1)=T9(N9+1,C8+1)*U9+T9(N9+1,C9+1)*(C1-U9)
1838 R1(C1+1)=28
1840 R1(C2+1)=100
1842 R1(C3+1)=10
1844 GOSUB 2218
1846 V1(Z2+1)=R6
1848 REM SEA STATE FUNCTION, F(SS)
1850 V1(C6+1)=C1+V1(C1+1)/D1(C6+1)
1852 REM VISIBILITY FUNCTION
1854 R1(Z2+1)=5:GOSUB 2294
1856 V1(C7+1)=V1(C6+1)*(C1+EXP(5-30*R9))
1858 RETURN
```

```
1860 REM SECTION B ******************
1862 N5(20)=Z2: REM NRBD
1864 N5(21)=Z2: REM NRBL
1866 FOR J1=Z2TO N5(12)-C1
1868 P=(L2-N5(20))/(N5(14)+N5(12)-J1)
1870 IF RND(C1)>=PTHEN 1878
1872 N5(20)=N5(20)+C1
1874 Y1(C3+1,Z2+1)=Y1(C3+1,Z2+1)+C1
1876 GOTO 1888
1878 GOSUB  2022:REM R.B. LAUNCH PROBABILITY
1880 IF RND(C1)>PTHEN 1886
1882 N5(21)=N5(21)+C1
1884 GOTO 1888
1886 Y1(6,Z2+1)=Y1(6,Z2+1)+C1
1888 NEXT J1
1890 IF N5(21)<=Z2THEN 1966
1892 N5(6)=N5(Z2+1)-N5(C1+1)-N5(C2+1)-N5(C3+1)-N5(C4+1)*N5(21)
1894 N5(13)=N5(21)*D1(C1+1)
1896 T5(C7+1)=D1(C4+1)/N5(21)
1898 T5(C1+1)=C5*D1(C4+1)*(C1-C1/N5(21))
1900 N5(18)=N5(14)-L2+N5(20)
1902 N5(19)=INT(L*N5(15)+C5)
1904 J2=INT(C5*N5(18)+C5)
1906 J4=N5(18)-J2
1908 GOSUB 1976:REM HIGH SIDE LAUNCH PROBABILITY
1910 J2=INT(J2*P+C5)
1912 GOSUB 2004:REM LOW SIDE LAUNCH PROBABLILTY
1914 J4=INT(J4*P+C5)
1916 N5(C6+1)=N5(16)*(J2+J4)
1918 GOSUB 2016:REM LIFERAFT LAUNCH PROBABILITY
1920 N5(C7+1)=N5(17)*INT(N5(19)*P+C5)
1922 N1=N5(6)-N5(C6+1)-N5(C7+1)
1924 IF N1>Z2THEN 1934
1926 N5(C8+1)=-N1
1928 N5(C9+1)=Z2
1930 S=Z2
1932 GOTO 1940
1934 REM R1 + R2 CREWMEN ABANDON SHIP IN L.B. & L.R. ?
1936 N5(C8+1)=Z2
1938 N5(C9+1)=N1
1940 IF N5(C9+1)>N5(13)THEN 1958
1942 S=N5(C9+1)
1944 R1(Z2+1)=C0(T0+1,C8+1):GOSUB 2258
1946 T5(Z2+1)=R6
1948 R1(Z2+1)=D1(C3+1):GOSUB 2258
1950 T5(C2+1)=R6
1952 R1(Z2+1)=C1:GOSUB 2258
1954 T5(C2+1)=T5(C2+1)+R6
1956 RETURN
1958 REM N MEN LOST. POINT Z **********
1960 J1=C7:Z1=N5(11)+N5(C9+1)-N5(13)
1962 GOSUB 1656
1964 GOTO 1972
1966 REM N MEN LOST. POINT X
1968 J1=C3:Z1=N5(11)
1970 GOSUB 1660
1972 E9=15
1974 RETURN
```

```
1976 REM PROBABILITY OF HIGH SIDE LAUNCH
1978 P=(30-T7)/15
1980 GOSUB 1994
1982 P1(C9+1)=P
1984 P=C1-V1(C1+1)/40
1986 GOSUB 1994
1988 P1(11)=P
1990 P=P1(C9+1)*P1(11)
1992 RETURN
1994 IF P<Z2THEN 1996:GOTO 1998
1996 P=Z2
1998 IF P>C1THEN 2000:GOTO 2002
2000 P=C1
2002 RETURN
2004 REM PROBABILITY OF LOW SIDE LAUNCH **********
2006 P=(40-T7)/25
2008 GOSUB 1994
2010 P1(12)=P
2012 P=P*P1(11)
2014 RETURN
2016 REM PROBABILITY OF LIFERAFT LAUNCH *********
2018 P=P1(11)
2020 RETURN
2022 REM PROBABILITY OF R.B. LAUNCH **********
2024 P=C1-C5*(V1(C1+1)/D1(C6+1))
2026 GOSUB 1994
2028 P1(13)=P
2030 IF RND(C1)>C5THEN 2036
2032 P=(40-T7)/(40-D1(C7+1))
2034 GOTO 2038
2036 P=(30-T7)/(30-D1(C7+1))
2038 GOSUB 1994
2040 P=P1(13)*P
2042 RETURN


1656 REM
1658 Y1(J1+1,Z2+1)=Y1(J1+1,Z2+1)+C1
1660 J3=J1+C1
1662 Y1(J3+1,Z2+1)=Y1(J3+1,Z2+1)+Z1
1664 IF Z1<Y1(J3+1,C1+1)THEN 1666:GOTO 1668
1666 Y1(J3+1,C1+1)=Z1
1668 IF Z1>Y1(J3+1,C2+1)THEN 1670:GOTO 1672
1670 Y1(J3+1,C2+1)=Z1
1672 RETURN
```

```
2044 REM SECTION C ***********************
2046 W=W+C1
2048 FOR I=N5(11)TO C1STEP -C1
2050 GOSUB 2164 :REM ADD DELTA T(PU)
2052 GOSUB 2242 :REM T(HYP)
2054 IF I=Z2THEN 2056:GOTO 2058
2056 RETURN
2058 P1(Z2+1)=N5(C1+1)/I
2060 R=RND(C1)
2062 IF R>P1(Z2+1)THEN 2074
2064 REM MAN KNOCKED INTO WATER (N(K))
2066 P=P1(14)
2068 T5(C6+1)=T5(Z2+1)+T5(C1+1)+T5(C2+1)*V1(C6+1)
2070 N5(C1+1)=N5(C1+1)-C1
2072 GOTO 2082
2074 REM MAN ISOLATED AND JUMPED INTO WATER (N(CRI))
2076 P=P1(15)
2078 T5(C6+1)=C5*(T5(Z2+1)+T5(C2+1)*V1(C6+1))+T5(C1+1)
2080 N5(C2+1)=N5(C2+1)-C1
2082 REM COMPLETE PICK-UP
2084 GOSUB 2270:REM T(S) DUE TO FLOTATION
2086 T9=T5(C3+1)
2088 IF R9<T9THEN 2090:GOTO 2092
2090 T9=R9
2092 IF T9>T5(C6+1)THEN 2096
2094 GOTO 2102
2096 REM SAVED
2098 GOSUB 2210:REM SAVED STATISTICS
2100 S=S+1
2102 IF N5(13)>STHEN 2150
2104 REM MANEUVER R.B.TO L.B.
2106 REM GET TIME FOR STANDARD MANEUVER
2108 R1(Z2+1)=T5(C8+1)
2110 GOSUB 2258
2112 T5(C1+1)=T5(C1+1)+R6*V1(C6+1)
2114 REM GET TIME TO TRANSFER 1 MAN
2116 R1(Z2+1)=T5(C9+1)
2118 GOSUB 2258
2120 IF RND(C1)<=P1(C4+1)THEN 2142
2122 IF S<=N5(C8+1)THEN 2142
2124 REM TRANSFER EXC CREWMEN TO L.B.
2126 T5(C1+1)=T5(C1+1)+N5(C8+1)*R6*V1(C6+1)
2128 S=S-N5(C8+1)
2130 N5(C8+1)=0
2132 IF S<N5(13)THEN 2150
2134 REM N MEN LOST POINT Z2
2136 J1=C9:Z1=I-C1
2138 GOSUB 1656
2140 GOTO 2152
2142 REM TRANSFER S CREWMEN TO L.B.
2144 T5(C1+1)=T5(C1+1)+S*R6*V1(C6+1)
2146 N5(C8+1)=N5(C8+1)-S
2148 S=0
2150 NEXT I
2152 X1(C3+1)=X1(C2+1)/X1(C1+1)
2154 IF X1(C1+1)=Z2 THEN 2156:GOTO 2160
2156 IF X1(C2+1)=Z2THEN 2158:GOTO 2160
2158 X1(C3+1)=C1
2160 GOSUB 2174:REM UPDATE SAVED STATISTICS
2162 RETURN
```

```
2164 REM ADD TIME TO PICK UP ONE MAN FROM WATER *******
2166 R1(Z2+1)=T5(8)
2168 GOSUB 2258:REM FRANK NICKELS' NORMAL DIST.
2170 T5(C1+1)=T5(C1+1)+R6*V1(C7+1)
2172 RETURN
2174 REM UPDATE CUMULATIVE 'SAVED' STATISTICS  ********
2176 IF Y1(C1+1,Z2+1)>=Z2THEN 2192
2178 FOR J3=C1TO C2
2180 J=J3+C1
2182 Y1(J3+1,Z2+1)=X1(J+1)
2184 Y1(J3+1,C1+1)=X1(J+1)
2186 Y1(J3+1,C2+1)=X1(J+1)
2188 NEXT J3
2190 RETURN
2192 FOR J3=C1TO C2
2194 J=J3+C1
2196 Y1(J3+1,Z2+1)=Y1(J3+1,Z2+1)+X1(J+1)
2198 IF X1(J+1)<Y1(J3+1,C1+1)THEN 2200:GOTO 2202
2200 Y1(J3+1,C1+1)=X1(J+1)
2202 IF X1(J+1)>Y1(J3+1,C2+1)THEN 2204:GOTO 2206
2204 Y1(J3+1,C2+1)=X1(J+1)
2206 NEXT J3
2208 RETURN
2210 REM GATHER SAVED STATISTICS *********
2212 X1(C2+1)=X1(C2+1)+C1
2216 RETURN
```

```
2218 REM NORMAL DISTRIBUTION
2220 R0=RND(C1)
2222 IF R0=Z2THEN 2224:GOTO 2226
2224 R0=L5
2226 R3=RND(C1)
2228 V=SQR(-C2*LOG(R0))*COS(P7*R3)
2230 R6=V*R1(4)+R1(Z2+1)
2232 IF R6<R1(C1+1)THEN 2234:GOTO 2236
2234 R6=R1(C1+1)
2236 IF R6>R1(C2+1)THEN 2238:GOTO 2240
2238 R6=R1(C2+1)
2240 RETURN
2242 REM SURVIVAL TIME (HYPOTHERMIA)    ************
2244 R1(C1+1)=Z0*(V1(Z2+1)*(S1+S2*V1(Z2+1)↑C2)-S3)
2246 R1(C2+1)=Z0*(V1(Z2+1)*((S4*V1(Z2+1)↑C2)-S5)+S6)
2248 R1(Z2+1)=C5*(R1(C1+1)+R1(C2+1))
2250 R1(4)=Z5*(R1(C2+1)-R1(C1+1))
2252 GOSUB 2218
2254 T5(4)=R6
2256 RETURN
2258 REM FRANK NICKELS' NORMAL DIST.  ***********
2260 R1(C1+1)=Z2
2262 R1(C2+1)=C2*R1(Z2+1)
2264 R1(C3+1)=C5*R1(Z2+1)
2266 GOSUB 2218
2268 RETURN
2270 REM COMPUTE T(S) DUE TO FLOTATION.***********
2272 IF RND(1)<=PTHEN 2280
2274 REM MAN NOT WEARING PFD --------------
2276 R1(Z2+1)=T5(6)
2278 GOTO 2284
2280 REM MAN WEARING PFD -----------
2282 R1(Z2+1)=T5(C4+1)
2284 GOSUB 2294 :REM RAYLEIGH DIST.
2286 RETURN
2288 REM UNIFORM DISTRIBUTION **************
2290 U9=R1(C1+1)+(R1(C2+1)-R1(C1+1))*RND(C1)
2292 RETURN
2294 REM RAYLEIGH DISTRIBUTION *************
2296 R0=RND(C1)
2298 IF R0=Z2THEN 2300:GOTO 2302
2300 R0=L5
2302 R9=R1(Z2+1)*SQR(-C2*LOG(R0))
2304 RETURN
```

```
1200 REM
1202 SELECT PRINT 215(160)
1204 PRINT
1206 PRINT TAB(31);A1$
1208 PRINT
1210 PRINT TAB(41);
1212 PRINT "SIMULATION SERIAL NUMBER AS";S9:PRINT :PRINT TAB(C9);T9$(N9+1):S8=14
1213 PRINT TAB(S8);"LBP =";T9(N9+1,Z2+1);" FEET"
1214 PRINT TAB(S8);"POB =";T9(N9+1,C1+1)
1215 PRINT TAB(S8);T9(N9+1,C2+1);" - ";T9(N9+1,C3+1); "MAN LIFEBOATS"
1216 PRINT TAB(S8);T9(N9+1,C4+1);" - ";T9(N9+1,6);" MAN LIFERAFTS"
1217 PRINT :PRINT TAB(C9);"NUMBER OF RESCUE BOATS =";D1(Z2+1)
1220 M7$="MEAN TIME TO  "
1221 PRINT TAB(C9);"RESCUE BOAT CAPACITY =";
1222 PRINT D1(C1+1);"+";D1(C2+1);"MAN CREW (EACH BOAT)":PRINT
1223 PRINT TAB(S8);M7$;" LAUNCH RESCUE BOAT =";
1224 PRINT D1(C3+1);"MINUTES"
1225 PRINT TAB(S8);M7$;" PERFORM STANDARD MANEUVER =";
1226 PRINT D1(6);"MIN."
1227 PRINT TAB(S8);M7$;" PICK UP ONE MAN FROM WATER =";
1228 PRINT D1(C4+1);"MIN."
1229 PRINT TAB(S8);"RATED HEEL ANGLE OF RESCUE BOAT DAVITS =";
1230 PRINT D1(C7+1);"DEGS."
1231 PRINT TAB(S8);"RATED SEA STATE CAPABILITY =";
1232 PRINT D1(C6+1);"FT.SIGN. WAVE HGT."
1234 PRINT
1235 PRINT TAB(C9);C0$(T0+1):PRINT :PRINT TAB(S8);"ON AVG.: ";
1236 P3=100:S7=S8+C9
1238 PRINT P3*C0(T0+1,C2+1);
1240 PRINT "% OF CREW KILLED BY CASUALTY"
1241 PRINT TAB(S7);C0(T0+1,C3+1);
1242 PRINT "CREWMAN KNOCKED OVERBOARD"
1243 PRINT TAB(S7);P3*C0(T0+1,C4+1);
1244 PRINT "% OF CREW ISOLATED FROM LIFEBOATS"
1245 PRINT TAB(S7);P3*C0(T0+1,6);
1246 PRINT "% OF LIFEBOATS DAMAGED"
1247 PRINT TAB(S7);C0(T0+1,C8+1);
1248 PRINT "MIN. FROM BEGINNING OF CASUALTY"
1249 PRINT TAB(S7+4);
1250 PRINT "TO BEGINNING OF ABANDON SHIP."
1252 IF T0<C1THEN 1256:IF T0>C2THEN 1256
1253 PRINT TAB(S7);C0(T0+1,C6+1);
1254 PRINT " FT. LENGTH OF COLLISION DAMAGE"
1256 IF T0<C2 THEN 1260:IF T0=C3 THEN 1260
1257 PRINT TAB(S7);C0(T0+1,C7+1);
1258 PRINT " DEGREES HEEL ANGLE"
1260 PRINT :PRINT
1261 PRINT TAB(C9);"RESULTS OF ";X1(Z2+1)+C1;" SIMULATIONS: "
1262 SELECT PRINT 005(80)
1264 RETURN
```

```
1266 REM
1268 SELECT PRINT 215(160)
1269 N=X1(Z2+1)+C1
1270 PRINT
1271 IF Y1(Z2+1,Z2+1)=Z2 THEN 1272:GOTO 1273
1272 V=-C1:GOTO 1274
1273 V=Y1(C1+1,Z2+1)/Y1(Z2+1,Z2+1):V=INT(1000*V+C5)/1000
1274 T1=35:T2=T1+C8:T3=T2+C8:T4=T3+C8:PRINT TAB(S8);"OVERALL FRACTION OF";
1276 PRINT TAB(T2);"*********"
1277 PRINT TAB(S8);"MEN IN WATER RESCUED";TAB(T2);
1280 IF V<Z2 THEN 1281 :GOTO 1282
1281 PRINT "* ---";TAB(T2+C7);"*":GOTO 1283
1282 PRINT "*";V;TAB(T2+C7);"*"
1283 PRINT TAB(S8);"BY RESCUE BOAT";TAB(T2);
1284 PRINT TAB(T2);"*********":PRINT
1285 PRINTUSING 1286:PRINTUSING 1287:M=N
1286 %                                                    NO.          MEAN       MIN.      MAX.
1287 %                                                    TIMES
1288 FOR I=Z2 TO C2
1290 PRINT
1292 READ A$
1294 IF M=Z2 THEN 1300
1296 IF Y1(I+1,Z2+1)<Z2 THEN 1297:GOTO 1298
1297 Y1(I+1,Z2+1)=Z2
1298 Q1=0:S=T2:Q1=Y1(I+1,Z2+1)/M
1300 Q2=0:S=T3:Q2=Y1(I+1,C1+1)
1302 Q3=0:S=T4:Q3=Y1(I+1,C2+1)
1303 PRINTUSING 1305,A$,M,Q1,Q2,Q3
1304 M=W
1305 %          ############################   #####       ####.##    ###.##    ###.##
1306 NEXT I
1308 PRINT
1310 FOR I=C4 TO 10 STEP C2
1311 Q4=0:Q5=0:Q6=0:Q7=0
1312 PRINT :READ A$:IF I<C8THEN 1319
1313 GOSUB 9010:READ A$:IF I=C8THEN 1319:GOSUB 9010
1314 READ A$
1319 Q4=Y1(I-C1+1,Z2+1)
1320 IF I=C4THEN 1321:IF I=C6THEN 1321:GOTO 1322
1321 GOSUB 9020:GOTO 1330
1322 IF Y1(I-C1+1,Z2+1)<=Z2 THEN 1323:GOTO 1324
1323 Q5=0:GOTO 1326
1324 S=T2:Q5=Y1(I+1,Z2+1)/Y1(I-C1+1,Z2+1)
1326 S=T3:Q6=Y1(I+1,C1+1)
1328 S=T4:Q7=Y1(I+1,C2+1)
1329 GOSUB 9020
1330 NEXT I:PRINT HEX(0C)
1332 SELECT PRINT 005(80):RETURN
```

```
1482 REM PRINT VARIABLES.
1484 DATA "NO. IN WATER, N"
1486 DATA "NO. RESCUED BY BOAT, NS"
1488 DATA "RATIO NS/N"
1490 DATA "RESCUE BOAT DAMAGED"
1492 DATA "R.B. LAUNCH FAILED"
1494 DATA "*NRBRB, ALL BOATS FILLED TO"
1496 DATA "CAPACITY BEFORE LAUNCH"
1498 DATA "*NRBRB, RESCUE BOAT & ALL"
1500 DATA "BOATS & RAFTS IN WATER"
1502 DATA "FILLED TO CAPACITY"




9000 SELECT PRINT 005(80):PRINT HEX(03):RETURN
9010 PRINTUSING 9011,A$:RETURN
9011 %          ################################
9020 GOSUB 9030:GOSUB 9040:PRINTUSING 9021,A$,Q4,Q5$,Q6$,Q7$:RETURN
9021 %          ############################   #####       #####       ####      ####
9030 CONVERT Q4TO Q4$,(#####):CONVERT Q5TO Q5$,(#####)
9031 CONVERT Q6TO Q6$,(#####):CONVERT Q7TO Q7$,(#####):RETURN
9040 IF Q5=0THEN 9041:RETURN
9041 Q5$=" ":Q6$=" ":Q7$=" ":RETURN
```

Appendix B: Annotated List of Variables and Arrays In Abandon Ship Simulation

1. Variables that remain unchanged throughout simulation

A1$ = String "ABANDON SHIP SIMULATION"

B5 = 999

Z0 = 60

Z0$ = String " CHARACTERISTICS"

Z5 = 0.25

T0 = Casualty Number - 1

C1 = 1

C2 = 2

C3 = 3

C4 = 4

C5 = 0.5

C6 = 6

C7 = 7

C8 = 8

C9 = 9

L5 = 9.5E-39

M = Integer part of S9

= Temporary variable in Output Program

M1 = 27

M8 = 8

M9 = 7

M9 = 7

M7$ = String "MEAN TIME TO "

N = Number of simulations

N3 = Primary Casualty Number

= C0(T0+1, 10)

N9 = Ship Number - 1

P3 = 100

P0 = π = 3.14159

P7 = 2π = 2*PI

S7 = 23

S8 = 14

S9 = Serial Number

S1 = 0.0024

S2 = 8.1E-6
S3 = 0.1
S4 = 5.9E-5
S5 = 0.101
S6 = 2.25
Z2 = 0

2. Variables that change (simulation variables)

E9 = Flag to control simulation
= 0 for normal completion of simulation
= 15 if rescue boat launch(es) fail(s) or all boats filled before launch
= -15 for no one will get into water for all simulations of a run (grounding or foundering)

I = General prupose index

I5 = Index to array DD for all characteristics

J = General purpose index

J2 = Number of lifeboats on high side successfully launched

J1 = Index to array MM$ for all characteristics

J4 = Number of lifeboats on low side successfully launched

J3 = Index into array YY

K = General purpose index

K1 = Index to count number of simulations; also used as index during override of ship and casualty characteristics

L = Fraction of lifeboats plus rescue boats(s) and liferafts destroyed; later subtracted from one (1) to be fraction not destroyed

L2 = Number of lifeboats plus rescue boat(s) destroyed

N1 = Number of people who must abandon ship in lifeboats, rescue boat(s), or liferafts

P = Probability (various)

R = Random number

R0 = First random number for normal distribution and randon number for Rayleigh distribution

R3 = Second random number for normal distribution

R6 = Random deviate from normal distribution

R9 = Random deviate from Rayleigh distribution

S = Number of people in rescue boat(s) (excluding crew); also temporary variable in Output Program

T7 = Heel angle

T9 = Survival time

U9 = Random deviate from uniform distribution

V = Random deviate from normal distribution with zero mean and unity (1) standard deviation; also temporary variable in Output Program

W = Tally of number of times rescue boat(s) get(s) deployed

Z1 = Number of people in water when rescue boat fails to recover all people in water

3. Arrays

Array CO

Note: CO has two dimensions. Nine (9) rows (first dimension) and ten (10) columns (second dimension). Each row contains the data for a casualty type. The identification of the columns appears below.

| Column Number | Variable | Description |
|---|---|---|
| 1 | $P_3$ | Probability that a collision results in a fire |
| 2 | $P_5$ | Probability that an explosion results in a fire |
| 3 | $\mu_{CR}$ | Mean fraction of people on board killed |
| 4 | $\mu_K$ | Mean number of people knocked (or blown) overboard |
| 5 | $\mu_C$ | Mean fraction of people on board isolated |
| 6 | $\mu_L$ | Mean fraction of life boats destroyed |
| 7 | $\mu_{LD}$ | Mean length of collision damage |
| 8 | $\mu_\theta$ | Mean heel angle |
| 9 | $\mu_{TC}$ | Mean time for the casualty to develop |
| 10 | NC | Primary casualty type number (1 - 6) |

Array CO$

Note: CO$, an array of strings, has the text for the nine casualty types.

| Element Number | Contents |
|---|---|
| 1 | "FIRE" |
| 2 | "COLLISION & FIRE" |
| 3 | "COLLISION & SINKING" |
| 4 | "EXPLOSION & FIRE" |
| 5 | "EXPLOSION & SINKING" |
| 6 | "STRUCTURAL FAILURE" |
| 7 | "GROUNDING" |
| 8 | "CAPSIZING" |
| 9 | "FOUNDERING" |

Array D1

Note: This array receives the replies to the questions; however responses to the ship type and casualty type get immediately transfered to arrays TS and CC respectively. The responses to rescue boat overlay the first nine elements of the ship type responses (if any); the responses to the casualty type questions extend the array from element nine (9), the last rescue boat response, to element sixteen, the last casualty type question (if any).

| Element Number | Variable | Description |
|---|---|---|
| 1 | $N_{RB}$ | Number of rescue boats |
| 2 | $C_{RB}$ | Capacity of rescue boat excluding the rescue boat crew |
| 3 | $N_{RBC}$ | Number in rescue boat crew |
| 4 | $\mu_{T_L}$ | Mean time to launch in minutes (does not include time to report) |
| 5 | MTTPU | Mean time to pick up one man from water |
| 6 | $\mu_{T_{SM}}$ | Mean time to perform a standard maneuver |
| 7 | $h_{WLIM}$ | Maximum significant wave height at which the rescue boat can safely launch |
| 8 | $\theta_{MAX}$ | Rated (maximum) heel angle of rescue boat davits |

Arrays M5$, M6, and M4

Note: These three arrays contain the text for the questions (rescue boat, ship, and casualty), MM$, the minimum acceptable reply, MN, and the maximum acceptable reply, MX. The text appears below; the replies are gathered into array DD.

| Element Number | Applicable Variable | Text |
|---|---|---|
| 1 | $N_{RB}$ | "NO. OF RESCUE BOATS (1 OR 2)" |
| 2 | $C_{RB}$ | "BOAT CAP. (EXC. CREW)" |
| 3 | $N_{RBC}$ | "NO. IN BOAT CREW" |
| 4 | $\mu_{TL}$ | "MEAN TIME TO LAUNCH (MIN.)" |
| 5 | MTTPU | "MEAN TIME TO PICK UP" |
| 6 | $\mu_{T_{SM}}$ | "STD. MANEUVER TIME (MIN.)" |
| 7 | $h_{WLIM}$ | "MAX. WAVE HEIGHT (FT.)" |
| 8 | $\theta_{MAX}$ | "MAX. HEEL ANGLE (DEG.)" |
| 9 | ... | "ENDURANCE TIME (HRS.)" |
| 10 | ... | "(1=YES; 0=NO)" |
| 11 | $L_S$ | "SHIP LENGTH" |
| 12 | $N_{CR}$ | "POB SHIP" |
| 13 | $N_{LB}$ | "NO. OF LIFEBOATS" |
| 14 | $C_{LB}$ | "CAPACITY OF EACH L.B." |
| 15 | $N_{LR}$ | "NO. OF LIFERAFTS" |
| 16 | $C_{LR}$ | "CAPACITY OF EACH L.R." |
| 17 | $\alpha_W$ | "MODE OF SIGN. WAVE HGTS.--MID-WINTER" |
| 18 | $\alpha_S$ | "MODE OF SIGN. WAVE HGTS.--MID-SUMMER" |
| 19 | $\mu_{T_W}$ | "MEAN WATER TEMP.--MID-WINTER" |
| 20 | $\mu_{T_S}$ | "MEAN WATER TEMP.--MID-SUMMER" |
| 21 | $P_{14}$ | "PROB. RESCUE VESSEL NEARBY" |
| 22 | $\mu_{CR}$ | "MEAN FRACTION OF POB KILLED" |
| 23 | $\mu_K$ | "MEAN NO. OF POB KNOCKED OVERBOARD" |
| 24 | $\mu_C$ | "MEAN FRACTION OF POB ISOLATED" |
| 25 | $\mu_L$ | "MEAN FRACTION OF LIFEBOATS DESTROYED" |
| 26 | $\mu_{LD}$ | "MEAN LENGTH OF DAMAGE" |
| 27 | $\mu_\theta$ | "MEAN HEEL ANGLE" |
| 28 | $\mu_{TC}$ | "MEAN TIME FOR CASUALTY TO DEVELOP" |

Array N5

| Element Number | Variable | Description |
|---|---|---|
| 1 | $N_{CR}$ | Number of people on board |
| 2 | $N_K$ | Number of people knocked (or blown) into water |
| 3 | $N_{CRI}$ | Number of people isolated from L.B. & L.R. |
| 4 | $N_{CRT}$ | Number of people killed |
| 5 | $N_{RBC}$ | Number in rescue boat crew |
| 6 | $N_{AS}$ | Number of people to abandon ship |
| 7 | $R_1$ | Number of people that can abandon ship in life boats |
| 8 | $R_2$ | Number of people that can abandon ship in life rafts. |
| 9 | EXC | Excess capacity in L.B. & L.R. |
| 10 | J | Number of people who attempt to abandon ship in rescue boat |
| 11 | N | Number of people in water |
| 12 | $N_{RB}$ | Number of rescue boats |
| 13 | $C_{RB}$ | Capacity of rescue boat |
| 14 | $N_{LB}$ | Number of life boats on ship |
| 15 | $N_{LR}$ | Number of life rafts on ship |
| 16 | $C_{LB}$ | Capacity of each life boat |
| 17 | $C_{LR}$ | Capacity of each life raft |
| 18 | $N_L$ | Number of life boats available |
| 19 | $N_R$ | Number of life rafts available |
| 20 | $N_{RB_D}$ | Number of rescue boats destroyed |
| 21 | $N_{RB_L}$ | Number of rescue boats launched |

Array P1

| Element Number | Variable | Description |
|---|---|---|
| 1 | $P_K$ | Probability that a given man in the water was knocked (or blown) into the water (as opposed to isolated) |
| 2 | ... | Not used |
| 3 | ... | Not used |
| 4 | $P_3$ | Probability that a collision results in a fire |
| 5 | $P_{14}$ | Probability that a rescue ship is nearby |
| 6 | $P_5$ | Probability that an explosion results in a fire |
| 7 | ... | Not used |
| 8 | ... | Not used |
| 9 | $P_8$ | Probability that a collision damages a life boat |
| 10 | $P_9$ | Probability (due to heel angle only) of successfull launch of high side L.B. |
| 11 | $P_{10}$ | Probability (due to wave height only) of successfull launch of L.B. or L.R. |
| 12 | $P_{11}$ | Probability (due to heel angle only) of successfull launch of low side L.B. |
| 13 | $P_{12}$ | Probability (due to wave height only) of successfull launch of rescue boat |
| 14 | $P_{13}$ | Probability man knocked (or blown) into water was wearing a PFD |
| 15 | $P_{14}$ | Probability man isolated & jumped into water was wearing a PFD |

RR→R1
Array R1

| | | |
|---|---|---|
| 1 | $\mu$ or $\alpha$ | Mean or Mode for Normal and Rayleigh Distributions |
| 2 | (No Name) | Minimum (Normal and Uniform Distributions) |
| 3 | (No Name) | Maximum (Normal and Uniform Distributions) |
| 4 | $\sigma$ | Standard Deviation for Normal Distribution |

Appendix B (Con't)

Array T9

Note: T9 has two dimensions. Eight (8) rows ($1^{st}$ dimension) and eleven (11) columns ($2^{nd}$ dimension). Each row contains the data for a ship type.

| Column Number | Variable | Description |
|---|---|---|
| 1 | $L_S$ | Ship length in feet |
| 2 | $N_{CR}$ | Number of people on board |
| 3 | $N_{LB}$ | Number of life boats on ship |
| 4 | $C_{LB}$ | Capacity of each life boat |
| 5 | $N_{LR}$ | Number of life rafts on ship |
| 6 | $C_{LB}$ | Capacity of each life raft |
| 7 | $\alpha W$ | Mode of significant wave heights--mid-winter |
| 8 | $\alpha S$ | Mode of significant wave heights--mid-summer |
| 9 | $\mu_{T_W}$ | Mean water temperature--mid-winter |
| 10 | $\mu_{T_S}$ | Mean water temperature--mid-summer |
| 11 | $P_{14}$ | Probability that a rescue vessel is nearby |

Array T9$

Note: T9$, an array of strings, has the text for the eight ship types.

| Element Number | Contents |
|---|---|
| 1 | "TANKER" |
| 2 | "CONTAINER" |
| 3 | "LNG" |
| 4 | "FISHBOAT" |
| 5 | "TUG" |
| 6 | "BARGE CARRIER (LASH)" |
| 7 | "GREAT LAKES BULKER" |
| 8 | "FERRY" |

Array T5

| Element Number | Variable | Description |
|---|---|---|
| 1 | $T_C$ | Time from casualty to abandon ship |
| 2 | $T_{PU}$ | Time from rescue boat launched to $i^{th}$ crewman retrieved from water |
| 3 | $T_L$ | Time from abandon ship to rescue boat launched |
| 4 | $T_{HYP}$ | Time to die due to hyperthermia |
| 5 | $T_{SF}$ | Mode of time to drown with PFD |
| 6 | $T_{SNF}$ | Mode of time to drown without PFD |
| 7 | $T_{IW}$ | Length of time man is in water |
| 8 | MTTPU | Mean time to pick up one man with rescue boat |
| 9 | $T_{SM}$ | Mean time to perform standard maneuver |
| 10 | $T_{TR}$ | Mean time to transfer a man from R.B. to L.B., L.R., or rescue vessel |

Array V1

| Element Number | Variable | Description |
|---|---|---|
| 1 | WT | Water temperature |
| 2 | $h_W$ | Wave height |
| 3 | $L_{LB}$ | Computed length of life boat |
| 4 | $L_{RB}$ | Computed length of rescue boat |
| 5 | $1/S_L$ | Reciprocal of ship length |
| 6 | $L_T$ | Total length of all L.B. & rescue boats |
| 7 | $F_{SS}$ | Sea state function $(1- h_W/h_{WLIM})$ |
| 8 | $F_{VIS}$ | Visibility function times sea state function |

Array X1

| Element Number | Variable | Description |
|---|---|---|
| 1 | $N_{SIM}-1$ | One less than the number of simulations |
| 2 | N | Number of people in the water |
| 3 | $N_S$ | Number of people rescued with rescue boat |
| 4 | $N_S/N$ | Ratio of number of people rescued with the rescue boat to the number of people in the water |

Array Y1

Note: Y1 has two dimensions. Eleven (11) rows (first dimension) and three (3) columns (second dimension). The first column (subscript of one) contains the running totals, the second column (subscript of two) contains the minimums, and the third column (three subscript) contains the maximums.

| Row Number | Variable | Description |
|---|---|---|
| 1 | N | Number of people in the water (running sum) |
| 2 | $N_S$ | Number of people rescued by the rescue boat (running sum) |
| 3 | $N_S/N$ | Running sum of the ratio |
| 4 | X | Number of times a rescue boat was damaged (with two rescue boats could be incremented twice in a simulation) No entries in column two or three |
| 5 | $N_{X,Y}$ | Number of people not rescued by rescue boat due to R.B. damage or launch failure. Not printed out |
| 6 | Y | Number of times a rescue boat launch failed (no entries in column 2 or 3) |
| 7 | ... | Not used |
| 8 | Z | Number of times all boats filled before launch (no entries in columns 2 or 3) |
| 9 | $N_Z$ | Number of people not rescued by rescue boat due to no capacity before launch |
| 10 | ZZ | Number of times all boats in water become filled (no entries in columns 2 or 3) |
| 11 | $N_{ZZ}$ | Number of people not rescued by rescue boat due to no capacity after launch |

APPENDIX C: Man Overboard Operator Program Listing

```
1     REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
2     REM INPUT PROGRAM FOR MAN OVERBOARD SIMULATION  23 FEB 1978
10    GOSUB 9200:REM HOME & ERASE TV.
22    DIM D1(15),D2(3)
23    DIM F(10)
24    DIM M1$(17)30,M6(17),M7(17):REM WWWWWWWWWWWWWWWWWWWWWWWWWWWW
25    DIM P2(15),R(4)
26    DIM T1(8,12),T1$(8)20,T3(10),T5(15):REM WWWWWWWWWWWWWWWWWWWWW
28    DIM V(6)
30    REM GET BEGINNING & ENDING SERIAL NUMBERS.  *********************
32    INPUT "FIRST SERIAL NUMBER";N5
34    INPUT "LAST SERIAL NUMBER";N6
40    INPUT "DATA FILE NAME";F$
42    SELECT #1 B20:DATA LOAD DC OPEN R#1,F$
68    FOR N3=N5 TO N6
70    REM INITIALIZE.  ******************************************************
72    PRINT:PRINT "INITIALIZING."
74    GOSUB 9000:REM INITIALIZE CONSTANTS.
76    GOSUB 9100:REM INITIALIZE ARRAYS.
78    GOSUB 9200:REM HOME & ERASE TV.
80    REM INPUT SHIP CHARACTERISTICS.  *********************************
82    PRINT "SHIP";C1$
84    I1=C1
86    PRINT "    TYPE OF SHIP?"
90    FOR I=C1 TO M5
100     PRINT TAB(4);I;". ";T1$(I)
110   NEXT I:PRINT
120   INPUT "NUMBER OF SHIP DESIRED";N1
130   IF N1<C1 THEN 80:IF N1>M5 THEN 80
140   GOSUB 9200:REM HOME & ERASE TV.
150   J1=M3:GOSUB 9300:REM STD. SHIP?
160   IF D1$="1" THEN 240
170   K1=C1
180   FOR J1=M3+C1 TO M4-C1
190     PRINT:PRINT"   (STD.";M1$(J1);" =";T1(N1,D2(K1));")";
200     GOSUB 9300:REM ASK QUESTION & RECEIVE REPLY.
210     T1(N1,D2(K1))=D1(K1)
220     K1=K1+C1
230   NEXT J1
240   P2(10)=T1(N1,11)
250   I1=C1
260   GOSUB 9200:REM HOME & ERASE TV.
275   REM INPUT RESCUE BOAT CHARACTERISTICS.  **************************
270   PRINT "RESCUE BOAT";C1$
280   PRINT:PRINT
290   FOR J1=C1 TO M3-C1
300     GOSUB 9300:REM ASK QUESTION & RECEIVE ANSWER.
310   NEXT J1
320   D1(I1)=C0
330   IF D1$<>"1" THEN 340
335   D1(I1)=C1
340   J1=M4:GOSUB 9300:REM ASK LAST R.B. QUESTION & RECEIVE ANSWER.
350   IF D1$<>"1" THEN 360
```

```
355    T1(N1,5)=C1
360    P2(5)=T1(N1,5):P2(6)=D1(12)
365    P2(15)=0.95+0.01*D1(7)
370    REM
372    REM # OF SIMULATIONS.  **********************************************
374    REM
378    INPUT "NUMBER OF SIMULATIONS";N4
380    REM
390    REM COMPUTE VARIABLES THAT REMAIN
400    REM CONSTANT FOR ALL SIMULATIONS IN A RUN.  ***********************
410    REM
414    T5(3)=10
418    T5(4)=30
420    T5(8)=T1(N1,3)*(C1-D1(9)/T1(N1,C2))
430    IF T5(8)>C0 THEN 440
435    T5(8)=C0
440    T5(13)=C2*C9*T1(N1,C1)/(76.05*T1(N1,C2))
445    T5(9)=300*T5(13)/180
450    T5(11)=15
460    T5(12)=30
500    REM WRITE VARIABLES ON DISK.  **************************************
510    OPEN "DUMMY" AS 2:READ #2;A$:CLOSE 2:REM  WWWWWWWWWWWWWWWWWWWWWWWW
520    DATA SAVE DC #1,N1,N2,N3,N4:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
540    FOR I=C1 TO M2
560      DATA SAVE DC #1,T1(N1,I),D1(I),P2(I),T5(I):REM WWWWWWWWWWWWWWWWWWWWW
580    NEXT I
600    FOR I=M3 TO 15
620      DATA SAVE DC #1,D1(I),P2(I),T5(I):REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
640    NEXT I
660    DATA SAVE DC #1,T1$(N1):REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
680    GOSUB 8600:REM DEBUG PRINT.  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
690    IF N3=N6 THEN 799
700    REM WANT MORE? ****************************************************
710    PRINT "WANT TO INPUT NEXT RUN?"
720    PRINT "      SERIAL # ";N3+C1
730    INPUT "      (1=YES; 0=NO)";A
740    IF A=C0 THEN 800
750    IF A<>C1 THEN 700
760    RESTORE
799    NEXT N3
800    PRINT "INPUT PHASE COMPLETE."
820    DATA SAVE DC #1,END:DATA SAVE DC CLOSEALL:STOP
8600   REM DEBUG PRINTOUTS.  *********************************************
8602   SELECT PRINT 215(160)
8610   PRINT:PRINT T1$(N1);
8615   T4=22
8620   FOR I2=C1 TO 12
8622     IF I2<>7 THEN 8626
8624     T4=22:PRINT
8626     PRINT TAB(T4);T1(N1,I2);
8630     T4=T4+9
8635   NEXT I2
8640   PRINT:PRINT:PRINT "RESCUE BOAT";
```

```
8645    T4=22
8650    FOR I2=C1 TO 12
8652      IF I2<>7 THEN 8654
8653      T4=22:PRINT
8654      PRINT TAB(T4);D1(I2);
8656      T4=T4+9
8658    NEXT I2
8660    PRINT:PRINT:PRINT "PROBABILITIES";
8662    T4=22
8664    FOR I2=C1 TO 15
8665      IF I2<>7 THEN 8666:GOTO 8667
8666      IF I2<>13 THEN 8668
8667      T4=22:PRINT
8668      PRINT TAB(T4);P2(I2);:T4=T4+9
8669    NEXT I2
8670    PRINT:PRINT:PRINT "S/N";TAB(13);"# OF SIM.";
8675    PRINT TAB(24);"T(SL)";TAB(42);"T(TW)";TAB(58);"T(LS(O))";
8680    PRINT TAB(67);"ALPHA(LS)"
8685    PRINT N3;TAB(13);N4;TAB(24);T5(8);TAB(42);T5(9);
8690    PRINT TAB(58);T5(11);TAB(67);T5(12)
8694    PRINT
8696    SELECT PRINT 005(80)
8699    RETURN
9000    REM INITIALIZE  ****************************************
9010    C0=0:C1=1:C2=2
9014    C3=11:C4=10
9016    C8=60:C9=3.14159
9020    C1$=" CHARACTERISTICS."
9025    D2(1)=1:D2(2)=2:D2(3)=12
9030    F1=254:G1=6
9040    M1=33:M2=12
9045    M3=13:M4=17:M5=8
9050    P3=0.9595:P4=0.0095
9060    P5=0.0168:P6=0.0008
9064    S1=0.0024:S2=8.1E-6:S3=0.1
9066    S4=5.9E-5:S5=0.101:S6=2.25
9080    X0=19
9099    RETURN
9100    REM SET UP ARRAYS.  ***********************************
9110    FOR I=C1 TO M4
9115      READ M1$(I),M6(I),M7(I)
9120    NEXT I
9125    FOR I=C1 TO M5
9130      READ T1$(I)
9135      FOR J=1 TO M2
9140        READ T1(I,J)
9145      NEXT J
9150    NEXT I
9160    FOR I=C1 TO 15
9170      READ P2(I)
9180    NEXT I
9199    RETURN
9200    REM HOME & ERASE TV.  **********************************
```

```
9210     PRINT HEX(03)
9220     RETURN
9300     REM DISPLAY QUESTION & RECEIVE ANSWER.  ****************
9310     PRINT:PRINT "   ";M1$(J1);
9315     IF M6(J1)=M7(J1) THEN 9370
9320     INPUT D1(I1)
9325     IF D1(I1)<M6(J1) THEN 9310
9326     IF D1(I1)>M7(J1) THEN 9310
9340     I1=I1+C1
9360     RETURN
9370     PRINT:PRINT "      (1=YES; 0=NO)";
9375     INPUT "ENTER 0 OR 1",D1$
9380     IF D1$="1" THEN 9399
9385     IF D1$<>"0" THEN 9310
9399     RETURN
9800     REM DATA STATEMENTS FOR QUESTIONS.  **************
9805     DATA "SPEED IN SMOOTH WATER (KNOTS)",3,100
9810     DATA "SPEED IN 8 FT. WAVES (KNOTS)",3,100
9815     DATA "FREEBOARD (FEET)",0.5,20
9820     DATA "PERCENT OF GUNWALE OPEN (%)",1,100
9825     DATA "HEIGHT OF LOOKOUTS EYE (FT)",3,50
9830     DATA "MEAN TIME TO PREPARE (MIN.)",0,10
9835     DATA "NUMBER OF FALLS",1,2
9840     DATA "DESCENT SPEED (FT./MIN.)",10,500
9845     DATA "MAX. LAUNCH SPEED (KNOTS)",0,100
9850     DATA "MAX. WAVE HEIGHT (FEET)",5,100
9855     DATA "STD. MANEUVER TIME (MIN.)",0,10
9860     DATA "USE BOAT FOR SEARCH",0,0
9865     DATA "WANT STANDARD SHIP",0,0
9870     DATA "SHIP LENGTH",10,2000
9875     DATA "SHIP SPEED",10,50
9880     DATA "MEAN HEADING ERROR (DEG.)",0,5
9885     DATA "SEARCH WITH R.B. ONLY",0,0
9900     REM DATA STATEMENTS FOR STANDARD SHIP CHARACTERISTICS.
9910     DATA "TANKER",845,16,8.5,50,0.05,3,6.5,10,70,38,0.05,1
9920     DATA "CONTAINER",710,23,2.6,60,0.05,3,5,24,70,38,0.05,1
9930     DATA "LNG",900,20,6.4,90,0.05,3,5,54,70,38,0.05,1
9940     DATA "FISHING",75,10,0.5,20,0.1,2.2,4.9,10,70,38,0.5,2
9950     DATA "TUG",150,7,8,25,0.5,2.2,4.9,10,70,38,0.5,2
9960     DATA "BARGE CARRIER (LASH)",740,21,6,60,0.05,3,5,25,70,38,0.05,1
9970     DATA "GREAT LAKES BULKER",708,14,9.4,37,0.2,2,4,13,60,42,0.05,1
9980     DATA "FERRY",150,16,0.5,35,0.15,3,5,8,70,38,0.5,2
9990     REM PROBABILITIES.
9992     DATA 0.63,0.12,0.03,0.26,0.0
9993     DATA 0.0,1.0,0.5,0.9,0.0
9994     DATA 0.8,0.9,0.0,0.65,0.0
```

Appendix D: Annotated List of Variables and Arrays In Man Overboard Simulation

1. Variables that remain unchanged throughout simulation

A$ = Input disk file name

B$ = Output disk file name in Simulation Program

= String of blanks in Output Program

B9 = 1.0E30 (a large value near the largest the machine can represent)

C0 = 0

C1 = 1

C1$ = string " CHARACTERISTICS."

C2 = 2

C3 = 11

C4 = 10

C8 = 60

C9 = 3.14159

F1 = 254 in Operator Program

= temporary storage for flag F(6) in Simulation Program

F9 = 0

G1 = 6

H$ = string " NO. MEAN MIN. MAX. "

L1 = 3

L2 = 8

L3 = 32 "TAB" values for "REGULAR PRINT"

L4 = 22

L5 = 41

L6 = 50

K$ = string "KNOTS"

M1 = 33

M2 = 12

M3 = 13

M4 = 17

M5 = 8

N1 = Ship Number

Appendix D (Con't)

N2 = 0

N3 = Serial Number

N4 = Number of Simulations

N5 = First Serial Number (Operator Program)

N6 = Last Serial Number (Operator Program)

P3 = 0.9595

P4 = 0.0095

P5 = 0.0168

P6 = 0.0008

(P3–P6: Constants to compute probability $P_{15}$)

S = Season (Time of Year)

S1 = 0.0024

S2 = 8.1E-6

S3 = 0.1

S4 = 5.9E-5

S5 = 0.101

S6 = 2.25

(S1–S6: Constants to compute minimum and maximum time to die due to hyperthermia)

T0 = Survival Time

X0 = 19

2. Variables that change (simulation variables)

F1 = value of flag F(6) after search by ship during search with both sequence

I = General purpose index

I1 = General purpose index

I2 = General purpose index

J = General purpose index

J1 = Index into Question Array, M1$

J7 = General purpose index

J8 = Index into results arrays, M9, N9, S9, & T9 for output

J9 = Index into results arrays, M9, N9, S9, & T9 for accumulating statistical results

K = Loop counter for search

K1 = value of K after search by ship during search with both sequence in Simulation Program

= Index into array D1 in Operator Program

Appendix D (Con't)

N = Overall fraction of MOB's rescued with rescue boat

P = Probability MOB has flotation (input to subroutine 8500)

R1 = First random number for distributions

R2 = Second random number for normal distribution

R3 = Variant selected from a distribution

T = Running Time (simulation time variable)

T2 = Time to find MOB including time to return (T4)

T4 = Time to return (from bridge notified to MOB's position)

T6 = Value of T2 after search with ship during search with both

T8 = Value of T4 after search with ship during search with both

V1 = Normal variant with mean of zero and unity standard deviation

3. Arrays

Array D0: An array of values originally used to truncate and format variables for output in lieu of PRINT USING statements. Not used in current version of program.

Array D1 Rescue Boat Characteristics

| Element Number | Variable | Description | Units |
|---|---|---|---|
| 1 | $V_{B_0}$ | Speed in smooth water | Knots |
| 2 | $V_{B_8}$ | Speed in 8 ft. sign. height waves | Knots |
| 3 | | Freeboard | Feet |
| 4 | | Percent of gunwale open (%) | ... |
| 5 | | Height of lookout's eye | Feet |
| 6 | | Mean time to prepare | Minutes |
| 7 | | Number of falls | ... |
| 8 | | Descent speed | Ft./Min. |
| 9 | | Maximum launch speed | Knots |
| 10 | | Maximum wave height | Knots |
| 11 | | Mean time to perform standard maneuver | Minutes |
| 12 | | Use boat for search (1=Yes; 0=No) | ... |

Appendix D (Con't)

Array D2 Index of ship characteristics to change

| Element Number | Value |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 12 |

Array D9 Print requested flags (1 = YES; 0 = NO)

| Element Number | Description |
|---|---|
| 1 | DEBUG PRINT # 1 flag |
| 2 | DEBUG PRINT # 2 flag |
| 3 | DEBUG PRINT # 3 flag |
| 4 | REGULAR PRINT flag |
| 5 | COMPRESSED PRINT flag |

Array F Flags set at strategic points in Simulation Program

| Element Number | Description |
|---|---|
| 1 | Seen - Not Seen (Seen = 2; Not Seen = 1) |
| 2 | Restricted - Not restricted (Restricted = 2; Not = 1) |
| 3 | Use rescue boat for search (NO = 1; Yes = 2) |
| 4 | Search results |
| 5 | Not used |
| 6 | Simulation results |

Arrays M1$, M6 and M7 Questions for operator and limits

| Element Number | Question (contents of M1$) | Minimum (M6) | Maximum (M7) |
|---|---|---|---|
| 1 | SPEED IN SMOOTH WATER (KNOTS) | 3 | 100 |
| 2 | SPEED IN 8 FT. WAVES (KNOTS) | 3 | 100 |
| 3 | FREEBOARD (FEET) | 0.5 | 20 |
| 4 | PERCENT OF GUNWALE OPEN (%) | 1 | 100 |
| 5 | HEIGHT OF LOOKOUTS EYE (FT) | 3 | 50 |

| | | | |
|---|---|---|---|
| 6 | MEAN TIME TO PREPARE (MIN.) | 0 | 10 |
| 7 | NUMBER OF FALLS | 1 | 2 |
| 8 | DESCENT SPEED (FT./MIN.) | 10 | 500 |
| 9 | MAX. LAUNCH SPEED (KNOTS) | 0 | 100 |
| 10 | MAX. WAVE HEIGHT (FEET) | 5 | 100 |
| 11 | STD. MANEUVER TIME (MIN.) | 0 | 10 |
| 12 | USE BOAT FOR SEARCH | 0 | 0 |
| 13 | WANT STANDARD SHIP | 0 | 0 |
| 14 | SHIP LENGTH | 10 | 2000 |
| 15 | SHIP SPEED | 10 | 50 |
| 16 | MEAN HEADING ERROR (DEG.) | 0 | 5 |
| 17 | SEARCH WITH R.B. ONLY | 0 | 0 |

Array M9 Simulation results (maximum time of occurance)

| Element Number | Description |
|---|---|
| 1 | MOB alive when bridge notified (seen) |
| 2 | MOB alive when ship or rescue boat returns to MOB's position (seen) |
| 3 | MOB alive when found in water (seen) |
| 4 | MOB alive when brought aboard rescue boat (or ship) (seen) |
| 5 | MOB alive when back aboard ship (seen) |
| 6 - 10 | Same as 1 - 5 but for when MOB not seen going overboard |

Array N9 Simulation results (minimum time of occurance)

Elements same as for array M9.

Array P2 Probabilities (Variable name same as element number)

| Element Number | Description |
|---|---|
| 1 | Probability MOB seen going overboard |
| 2 | Probability MOB had flotation when seen going overboard |
| 3 | Probability MOB was wearing a PFD when not seen going overboard |
| 4 | Probability MOB got a Life Ring when seen going overboard |

Appendix D (Con't)

| | |
|---|---|
| 5 | Probability ship restricted from turning (set to unity for search with rescue boat only) |
| 6 | Probability both rescue boat and ship will search (set to unity for "use rescue boat for search") |
| 7 | Not used |
| 8 | Probability MOB is spotted in water during a given pass when MOB not seen going overboard |
| 9 | Not used |
| 10 | Probability ship will recover MOB directly when found by ship |
| 11 | Probability ship successfully recovers MOB when found by ship and ship elects to recover MOB directly |
| 12 | Same as $P_8$ but for cases when MOB is seen going overboard |
| 13 | Probability MOB and rescue boat crew will come aboard the ship and leave the rescue boat adrift |
| 14 | Probability MOB and rescue boat crew will successfully get back aboard ship when rescue boat is cast adrift |
| 15 | Probability rescue boat with MOB and rescue boat crew are successfully recovered along with rescue boat |

Array R Parameters for distribution subroutines

| Element Number | Description |
|---|---|
| 1 | Mean for normal distributions and mode for Rayleigh distribution |
| 2 | Minimum for normal distribution and uniform distribution |
| 3 | Maximum for normal and uniform distributions |
| 4 | Standard deviation for normal distribution |

Array S8 Result of simulation which require only a count of the number of times the event occurs

| Element Number | Description |
|---|---|
| 1 | MOB seen going overboard |
| 2 | MOB not seen going overboard |

3 Rescue boat launch failed due to sea state

4 Rescue boat successfully deployed

Array S9 Simulation results (number of successful rescues using rescue boat)
Elements same as for array M9.

Array T1 Ship characteristics

Note: In the Operator Program this array has two dimensions (Indexes). The first index selects the ship type; the second selects the ship characteristic. In the Simulation and Output Programs the single index selects the ship characteristic. The changing from two dimensions to one dimension occurs as a result of the separate programs with a disk file serving as temporary storage.

Array T1 Ship Characteristics

| Element Number | Variable | Description | Units |
|---|---|---|---|
| 1 | $L_S$ | Ship length | Feet |
| 2 | $V_S$ | Ship speed | Knots |
| 3 | $T_{SL_O}$ | Time to stop | Minutes |
| 4 | | Height of lookout's eye | Feet |
| 5 | $P_5$ | Probability ship is restricted from turning | ... |
| 6 | | Mode of sign. wave heights in mid-summer | Feet |
| 7 | | Mode of sign. wave heights in mid-winter | Feet |
| 8 | | Freeboard of ship | Feet |
| 9 | | Mean water temperature in mid-summer | Deg. F. |
| 10 | | Mean water temperature in mid-winter | Deg. F. |
| 11 | $P_{10}$ | Probability MOB will be recovered by ship rather than rescue boat | ... |
| 12 | | Mean heading error | Deg. |

Appendix D (Con't)

Array T1$ Ship type used only by Operator Program (a variable in the Simulation and Output Programs holds the ship type)

| Element Number | Contents |
|---|---|
| 1 | TANKER |
| 2 | CONTAINER |
| 3 | LNG |
| 4 | FISHING |
| 5 | TUG |
| 6 | BARGE CARRIER (LASH) |
| 7 | GREAT LAKES BULKER |
| 8 | FERRY |

Array T3 Parameters for search (also see TABLE I in body of report)

| Element Number | Description |
|---|---|
| 1 | Time for ship to turn or time to prepare to launch rescue boat |
| 2 | Time for ship or rescue boat to return to MOB's position |
| 3 | Survival time remaining at beginning of search |
| 4 | Navigational Error |
| 5 | Distance of visibility |
| 6 | Time since MOB last seen |
| 7 | Time to turn (180 deg. turn) |
| 8 | Time from time MOB went overboard until MOB missed |
| 9 | Number of opportunities to spot MOB during search until survival time expended |
| 10 | Probability MOB will be spotted in water during a given pass |

Array T5 Various values that remain constant throughout a simulation

| Element Number | Description |
|---|---|
| 1 | Time since MOB last seen |
| 2 | Time from when MOB went overboard until MOB missed |

| | |
|---|---|
| 3 | Mode of survival time due to flotation without flotation |
| 4 | Mode of survival time due to flotation with flotation |
| 5 | Time to notify bridge |
| 6 | Time for rescue boat to report to boat |
| 7 | Time to prepare rescue boat for launch |
| 8 | Time for ship to slow to the maximum safe rescue boat launch speed |
| 9 | Time for ship to execute a Williamson turn |
| 10 | Survival time |
| 11 | Minimum time since last seen |
| 12 | Mode of time since last seen minus minimum time since last seen |
| 13 | Time for ship to execute a 180 degree turn |
| 14 | Time to launch rescue boat |
| 15 | Environmental distance of visibility |

Array T9 Simulations results (sum of times of occurance of events)

Note: This array holds the running total time for the event during the simulation run. After the simulation run each element of this array gets divided by the corresponding element of array S9 (number of events) to yield the mean time for the event.

Elements same as for array M9.

Array V Various variables

| Element Number | Description |
|---|---|
| 1 | Water temperature |
| 2 | Wave height |
| 3 | Rescue boat speed (in the height of waves in 2) |
| 4 | Speed ratio (Ship speed divided by boat speed) |
| 5 | Distance of visibility from ship |
| 6 | Distance of visibility from rescue boat |

# APPENDIX E: Man Overboard Simulation Program Listing

```
1    REM PROGRAM FOR MAN OVERBOARD SIMULATION           9 MARCH 1978
10   GOSUB 9200:REM HOME & ERASE TV.
20   PRINT "INITIALIZING."
22   DIM D0(10),D1(15),D9(5)
23   DIM F(6)
24   DIM M1$(17)30,M6(17),M7(17),M9(10) REM WWWWWWWWWWWWWWWWWWWWWWWWW
25   DIM N9(10),P2(15),R(4),S8(4),S9(11)
26   DIM T1(12),T3(10),T5(15),T9(10)      REM WWWWWWWWWWWWWWWWWWWWWWW
28   DIM V(6)
30   GOSUB 9000:REM INITIALIZE (CONSTANTS).
40   INPUT "DEBUG PRINT #1 (1=YES; 0=NO)";D9(C1)
42   IF D9(C1)<>C0 THEN 43:GOTO 50
43   IF D9(C1)<>C1 THEN 40
50   INPUT "DEBUG PRINT #2 (1=YES; 0=NO)";D9(C2)
52   IF D9(C2)<>C0 THEN 53:GOTO 54
53   IF D9(C2)<>C1 THEN 50
54   INPUT "DEBUG PRINT #3 (1=YES; 0=NO)";D9(3)
56   IF D9(3)<>C0 THEN 57:GOTO 60
57   IF D9(3)<>C1 THEN 54
60   INPUT "REGULAR PRINT (1=YES; 0=NO)";D9(4)
62   IF D9(4)<>C0 THEN 63:GOTO 66
63   IF D9(4)<>C1 THEN 60
66   INPUT "COMPRESSED PRINT (1=YES; 0=NO)";D9(5)
68   IF D9(5)<>C0 THEN 69:GOTO 80
69   IF D9(5)<>C1 THEN 66
80   REM RANDOMIZE:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
90   INPUT "INPUT DATA FILE NAME";A$
92   IF D9(5)<C1 THEN 100
94   REM LPRINTER:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
96   GOSUB 7900:REM COMPRESSED PRINT-OUT HEADING.
98   REM CONSOLE:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
100  REM READ VARIABLES FROM DISK.  *********************************
118  Z$=A$
120  SELECT #1 B20:DATA LOAD DC OPEN R#1,A$
140  REM BEGIN LOOP TO DO RUNS TO END-OF-FILE.  *********************
150  DATA LOAD DC #1,N1,N2,N3,N4:IF END THEN 400:REM WWWWWWWWWWWWWWWW
152  IF A$<>Z$ THEN 158
156  SELECT #2 B20:DATA LOAD DC OPEN R#2,B$
160  DATA SAVE DC #2,N1,N2,N3,N4:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
170  FOR I=C1 TO M2
180    DATA LOAD DC #1,T1(I),D1(I),P2(I),T5(I):REM WWWWWWWWWWWWWWWWWW
185    DATA SAVE DC #2,T1(I),D1(I),P2(I),T5(I):REM  WWWWWWWWWWWWWWWWW
190  NEXT I
200  FOR I=M3 TO 15
220    DATA LOAD DC #1,D1(I),P2(I),T5(I):REM WWWWWWWWWWWWWWWWWWWWWWWW
230    DATA SAVE DC #2,D1(I),P2(I),T5(I):REM  WWWWWWWWWWWWWWWWWWWWWWW
240  NEXT I
250  A$=B$
260  DATA LOAD DC #1,T1$:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
262  DATA SAVE DC #2,T1$:REM  WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
266  IF D9(3)<C1 THEN 300
270  REM LPRINTER:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
280  GOSUB 7600:REM DEBUG #3 PRINT-OUT.
```

```
285    REM CONSOLE:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
300    FOR I=C1 TO 10
320      M9(I)=-B9:N9(I)=B9
330      S9(I)=C0:T9(I)=C0
340    NEXT I
350    S8(C1)=C0:S8(C2)=C0:S8(3)=C0:S8(4)=C0
360    S9(11)=C0
399    GOTO 470
400    PRINT:PRINT "FINISHED.":DATA SAVE DC #2,END
410    DATA SAVE DC CLOSEALL
430    STOP
440    PRINT "OUTPUT DISK ERROR."
442    STOP
450    PRINT "CAN'T OPEN INPUT FILE."
452    STOP
460    PRINT "CAN'T OPEN OUTPUT FILE."
462    STOP
470    REM
472    REM BEGIN LOOP TO PERFORM N4 SIMULATIONS.  *********************
474    REM
480    FOR N8=C1 TO N4
490    REM
492    REM RESET FLAGS.  **************************************
494    REM
496    FOR I=C1 TO 6
497      F(I)=C0
498    NEXT I
499    F9=C0
500    REM
510    REM COMPUTE VARIABLES THAT REMAIN CONSTANT
520    REM FOR A GIVEN SIMULATION.  ***********************************
530    REM
550    REM
560    REM COMPUTE NAVIGATIONAL ERROR = N.E.  *****************
570    REM
580    R(C1)=C0
590    R(3)=C2*T1(C1):R(C2)=-R(3)
600    R(4)=T1(C1)
610    GOSUB 8000:REM CONSTANT TERM IN N.E.
620    N2=R3
625    REM CONTINUED LATER (LINE 922).
630    REM
635    REM SET SEASON (TIME OF YEAR).  ************************
640    REM
650    S=RND(1)
660    REM
664    REM ENVIRONMENTAL VISIBILITY.  *************************
666    REM
668    R(C1)=25000:GOSUB 8200:T5(15)=R3
670    REM
672    REM DISTANCE OF VISIBILITY FROM SHIP AND BOAT.
674    REM
676    GOSUB 9500
```

AD-A070 893 CASDE CORP TORRANCE CA F/G 5/2

PROGRAMMER'S MANUAL FOR COMPUTER PROGRAMS INVOLVED IN A STUDY O--ETC(U)

MAY 78 L B BROWN, R M DIJULIO, F J NICKELS DOT-CG-61276-A

UNCLASSIFIED USCG-D-9-78 NL

2 OF 2

AD A070893

END

DATE FILMED

8 --79

DDC

```
680     REM
682     REM WATER TEMPERATURE.  ********************************
684     REM
690     R(C1)=T1(9)+(T1(10)-T1(9))*S
692     R(C2)=28:REM MIN TEMP.
694     R(3)=100:R(4)=10
696     GOSUB 8000
698     V(C1)=R3
700     REM
710     REM SURVIVAL TIME (HYPERTHERMIA).  ********************
720     REM
730     GOSUB 8400:T5(10)=R3
740     REM
742     REM WAVE HEIGHT.  ****************************************
744     REM
750     R(C1)=T1(6)+(T1(7)-T1(6))*S
752     GOSUB 8200
754     V(C2)=R3
760     REM
762     REM PROBABILITY OF SUCCESSFUL RECOVERY OF R.B.  ********
764     REM
766     P2(15)=P3-P4*D1(7)-V(C2)*(P5-P6*D1(7))
770     REM
772     REM BOAT SPEED & SPEED RATIO.  **************************
774     REM
776     V(3)=D1(C1)+(D1(C2)-D1(C1))*V(C2)/8
780     IF V(3)>3 THEN 784
782     V(3)=3
784     V(4)=T1(C2)/V(3)
786     IF D9(C1)<C1 THEN 800
790     GOSUB 8700:REM DEBUG PRINT.
800     REM
810     REM LOGIC TO DETERMINE T(M) & FLOATATION.  ********************
820     REM
830     IF RND(1)<=P2(C1) THEN 1600
840     REM NOT SEEN GOING OVERBOARD.  ******************************
850     F(C1)=C1
855     S8(C2)=S8(C2)+C1
860     P=P2(3)
870     GOSUB 8500:REM SURVIVAL TIME (FLOTATION).
880     IF R3>T5(10) THEN 890
882     T5(10)=R3
888     REM COMPUTE TIME LAST SEEN & TIME MISSED (NOT SEEN).  **********
890     R(C1)=T5(12)
900     GOSUB 8200:REM RAYLEIGH DIST.
910     T5(C1)=T5(11)+R3:REM TIME LAST SEEN
920     T5(C2)=T5(C1)*RND(1):REM TIME MISSED.
922     REM COMPLETE N.E.
924     R(C1)=1.77*T1(12)*T5(C2)*T1(C2)
926     GOSUB 8100:REM SLOPE TERM IN N.E.
928     N2=ABS(N2+R3)
930     T=T5(C2)
932     REM SET TO TO SURVIVAL TIME.
```

```
934     TO=T5(10)
939     REM IS MAN STILL ALIVE (MISSED)?
940     IF T>=TO THEN 3100
945     REM COMPUTE TIME TO NOTIFY BRIDGE (NOT SEEN).  *****************
950     R(C1)=C1
960     GOSUB 8100:REM TIME TO NOTIFY BRIDGE.
970     T5(5)=R3
980     T=T+R3
989     REM IS MAN STILL ALIVE (BRIDGE NOTIFIED)?
990     IF T>=TO THEN 3200
1000    REM
1010    REM LOGIC TO HANDLE SEARCH BY BOAT, BY SHIP, OR BY BOTH. *******
1020    REM
1025    J9=6:GOSUB 6500
1030    IF RND(1)<=P2(5) THEN 1500
1040    REM SHIP NOT RESTRICTED. ***************************************
1050    F(C2)=C1
1060    IF P2(6)>C0 THEN 1200
1100    REM SEARCH WITH SHIP ONLY WHEN MOB NOT SEEN. *******************
1110    F(3)=C1
1120    Z=3
1130    GOSUB 7000:REM SET UP T3 ARRAY.
1140    GOSUB 9400:REM DO SEARCH.
1144    REM INCREMENT TIME.
1148    T=T+T4
1150    REM TEST FOR SUCCESS.
1160    IF F(4)<8 THEN 3300
1170    REM SUCCEEDED.
1174    J9=7:GOSUB 6500:REM REACHED MOB'S POSITION.
1180    T=T+T2-T4
1185    J9=8:GOSUB 6500:REM FOUND MOB.
1190    GOTO 1400
1200    REM SEARCH WITH BOTH WHEN MOB NOT SEEN. ************************
1210    REM FIRST WITH SHIP.
1212    F(3)=C2:Z=3
1214    GOSUB 7000:REM SET UP T3 ARRAY.
1216    T3(4)=0.5*T3(4):REM ADJ. N.E.
1218    GOSUB 9400:REM DO SEARCH.
1220    REM SAVE RESULTS.
1222    F1=F(4):K1=K:T6=T2:T8=T4
1230    REM THEN WITH BOAT.
1232    REM CAN LAUNCH?
1234    IF D1(10)>=V(C2) THEN 1240
1236    F(4)=4:T2=C0:T4=C0:S8(3)=S8(3)+C1
1238    F(3)=C1:GOTO 1250
1240    Z=4:S8(4)=S8(4)+C1
1242    GOSUB 7000:REM SET UP T3 ARRAY.
1244    T3(4)=0.5*T3(4):REM ADJ. N.E.
1246    GOSUB 9400:REM DO SEARCH.
1248    REM DETERMINE SUCCESS? & WHICH?
1250    IF F1>7 THEN 1256
1252    IF F(4)<8 THEN 3400
1254    GOTO 1260
```

```
1256    IF F(4)<8 THEN 1270:IF T6<T2 THEN 1270
1258    REM FOUND BY BOAT ONLY OR FOUND BY BOAT FIRST (OR AT SAME TIME).
1260    F(5)=4:T=T+T4
1262    J9=7:GOSUB 6500:REM REACHED MOB'S POSITION.
1264    T=T+T2-T4
1266    J9=8:GOSUB 6500:REM FOUND MOB.
1268    GOTO 2350:REM FOUND BY BOAT.
1269    REM FOUND BY SHIP ONLY OR FOUND BY SHIP FIRST.
1270    F(5)=3:T=T+T8
1272    J9=7:GOSUB 6500:REM REACHED MOB'S POSITION.
1274    T=T+T6-T8
1276    J9=8:GOSUB 6500:REM FOUND MOB.
1400    REM FOUND BY SHIP. *****************************************************
1410    REM MANEUVER NEAR MOB.
1420    R(C1)=C2:GOSUB 8100
1430    T=T+R3
1440    IF T>=T0 THEN 4900
1450    GOTO 2100
1500    REM SEARCH WITH BOAT WHEN MOB NOT SEEN. *************************
1510    Z=4
1520    GOTO 1820
1600    REM SEEN GOING OVERBOARD.  ************************************
1610    F(C1)=C2
1614    S8(C1)=S8(C1)+C1
1616    P=P2(C2)
1630    GOSUB 8500:REM SURVIVAL TIME (FLOATION).
1640    IF R3>T5(10) THEN 1660
1650    T5(10)=R3
1655    REM COMPUTE TIME TO NOTIFY BRIDGE (SEEN).  *********************
1660    R(C1)=C1
1670    GOSUB 8100:REM TIME TO NOTIFY BRIDGE.
1680    T5(5)=R3
1690    T=R3
1695    T0=T5(10)
1700    REM IS MAN STILL ALIVE (BRIDGE NOTIFIED).
1710    IF T>=T0 THEN 3600
1715    J9=C1:GOSUB 6500
1720    REM
1730    REM LOGIC TO HANDLE SEARCH BY BOAT OR SHIP (SEEN).  ************
1740    REM
1750    IF RND(1)>P2(5) THEN 2000
1800    REM SHIP IN RESTRICTED WATERS. *********************************
1810    Z=2
1820    F(C2)=C2
1830    REM CAN LAUNCH BOAT?
1840    IF D1(10)>=V(C2) THEN 1870
1850    F(4)=4:T2=C0
1860    GOTO 3700:REM FAILED NO LAUNCH.
1870    REM SET UP AND DO SEARCH.
1880    GOSUB 7000:REM SET UP T3 ARRAY.
1890    GOSUB 9400:REM DO SEARCH.
1895    T=T+T4
1900    REM DETERMINE SUCCESS.
```

```
1905    F(3)=C2:S8(4)=S8(4)+C1
1910    IF F(4)<8 THEN 3800
1930    J9=C2
1960    GOSUB 6500:REM REACHED MOB'S POSITION.
1970    J9=J9+C1
1980    T=T+T2-T4
1990    GOSUB 6500:REM FOUND MOB.
1999    GOTO 2350
2000    REM SHIP CAN TURN (SEEN).  ****************************************
2010    F(C2)=C1:F(3)=C1
2020    Z=1
2030    GOSUB 7000:REM SET UP T3 ARRAY.
2040    GOSUB 9400:REM DO SEARCH.
2045    T=T+T4
2050    REM DETERMINE SUCCESS.
2060    IF F(4)<8 THEN 3900
2075    J9=C2:GOSUB 6500:REM REACHED MOB'S POSITION.
2080    T=T+T2-T4
2085    J9=3:GOSUB 6500:REM FOUND MOB.
2100    REM RECOVER WITH SHIP?  *****************************************
2110    IF RND(1)>P2(10) THEN 2200
2120    REM USE SHIP FOR RECOVERY.
2130    IF RND(1)>P2(11) THEN 4100
2140    GOTO 5000
2200    REM MAN FOUND BY SHIP RECOVER WITH R.B.  ************************
2210    REM
2220    IF F(3)<C2 THEN 2300
2230    REM BOAT ALREADY LAUNCHED (SEARCHED WITH BOTH).
2235    IF K1=C1 THEN 2236:GOTO 2240
2236    IF F1=9 THEN 2260
2240    REM NOT FOUND ON FIRST PASS.
2245    R(C1)=0.25*V(4)*T5(C1)
2250    GOSUB 8100
2255    GOTO 2280
2260    REM FOUND ON FIRST PASS.
2270    R3=T5(C2)*ABS(V(4)-C1)
2280    T=T+R3
2285    IF T>=T0 THEN 4200
2290    GOTO 2350
2300    REM LAUNCH RESCUE BOAT.  ****************************************
2305    REM CAN LAUNCH BOAT?
2310    IF D1(10)<V(C2) THEN 4300
2320    REM LAUNCH RESCUE BOAT.
2330    T=T+T5(14)
2335    S8(4)=S8(4)+C1
2340    IF T>=T0 THEN 4400
2350    REM MANEUVER THE RESCUE BOAT TO RECOVER MOB.  ******************
2360    R(C1)=D1(11)
2370    GOSUB 8100
2380    T=T+R3
2390    IF T>=T0 THEN 4500
2400    REM RECOVER MOB. ************************************************
2410    R(C2)=D1(3)-C2
```

```
2420    R(C1)=(0.5+0.125*R(C2)*R(C2))*(C1+EXP(4-0.1*D1(4)))
2430    GOSUB 8100
2440    T=T+R3
2450    IF T>=TO THEN 4600
2460    J9=4
2490    GOSUB 6500:REM MOB IN R.B.
2500    REM UNLOAD OR RECOVER RESCUE BOAT.  *******************
2510    REM
2520    IF RND(1)<P2(13) THEN 2560
2530    REM UNLOAD R.B. CREW & MOB.  ***************************
2540    IF RND(1)<=P2(14) THEN 5100
2550    GOTO 4700
2560    IF RND(1)<=P2(15) THEN 5200
2570    GOTO 4800
3100    REM!! "DIED, MISSED.  T,TS -";T;TO
3120    F(6)=C1
3199    GOTO 5900
3200    REM!! "DIED, BRIDGE NOTIFIED.  T,TS =";T;TO
3220    F(6)=C2
3299    GOTO 5900
3300    REM!! "DIED, SEARCH FAILED.  T,TS,F(4) =";T;TO;F(4)
3320    F(6)=3
3330    IF F(4)=C1 THEN 5900
3340    J9=7:GOSUB 6500:REM REACHED MOB'S POSITION.
3399    GOTO 5900
3400    REM!! "DIED, SEARCH BY BOTH FAILED.  T,TS,F(4) =";T;TO;F(4)
3420    F(6)=4
3430    IF F1=C1 THEN 3440
3434    IF F(4)=C1 THEN 3450
3438    IF T4>T8 THEN 3450
3440    IF F(4)=C1 THEN 5900
3445    T=T+T4:GOTO 3460
3450    T=T+T8
3460    J9=C2:GOSUB 6500:REM REACHED MOB'S POSITION.
3499    GOTO 5900
3500    REM!! "DIED, RECOVERY BY SHIP FAILED.  T,TS =";T;TO
3520    F(6)=5
3599    GOTO 5900
3600    REM!! "DIED, BRIDGE NOTIFIED (SEEN).  T,TS =";T;TO
3620    F(6)=6
3699    GOTO 5900
3700    REM!! "R.B. LAUNCH FAILED.  W.H., F(4), T =";V(C2);F(4);T
3720    F(6)=7
3730    S8(3)=S8(3)+C1
3799    GOTO 5900
3800    REM!! "SEARCH FAILED (SEEN).  F(4),T =";F(4);T+T2
3820    F(6)=8
3830    IF F(4)=C1 THEN 5900
3840    J9=C2:GOSUB 6500:REM REACHED MOB'S POSITION.
3899    GOTO 5900
3900    REM!! "SEARCH FAILED (SEEN; SHIP).  F(4),T,TS =";F(4);T+T2;TO
3920    F(6)=9
3930    IF F(4)=C1 THEN 5900
```

```
3940   J9=C2:GOSUB 6500:REM REACHED MOB'S POSITION.
3999   GOTO 5900
4100   REM!! "RECOVERY WITH SHIP FAILED.  T ="$T
4120   F(6)=10
4199   GOTO 5900
4200   REM!! "DIED, BOAT RETURNING TO MOB POSITION.  T,TS ="$T$TO
4220   F(6)=11
4299   GOTO 5900
4300   REM!! "DIED, FOUND MOB BUT COULD NOT LAUNCH R.B.  WAVE HEIGHT ="$V(C2)
4320   F(6)=12
4330   S8(3)=S8(3)+C1
4399   GOTO 5900
4400   REM!! "DIED WHILE LAUNCHING R.B.  T,TS ="$T$TO
4420   F(6)=13
4499   GOTO 5900
4500   REM!! "DIED WHILE PERFORMING A STD. MAN.  T,TMB,TS ="$T$R3$TO
4520   F(6)=14
4599   GOTO 5900
4600   REM!! "DIED WHILE BRINGING ABOARD R.B.  T,TREM,TS ="$T$R3$TO
4620   F(6)=15
4699   GOTO 5900
4700   REM!! "LOST UNLOADING R.B.  T,TS ="$T$TO
4720   F(6)=16
4799   GOTO 5900
4800   REM!! "LOST R.B., CREW, AND MOB.  T,TS ="$T$TO
4820   F(6)=17
4899   GOTO 5900
4900   REM!! "DIED DURING MANEUVER SHIP NEAR MOB.  T,TS ="$T$TO
4920   F(6)=18
4999   GOTO 5900
5000   REM RECOVERED MAN WITH SHIP.
5020   F(6)=20
5099   GOTO 5270
5100   REM RECOVERED MAN FROM RESCUE BOAT.
5120   F(6)=21
5199   GOTO 5230
5200   REM RECOVERED RESCUE BOAT WITH MOB & CREW.
5220   F(6)=22
5230   J9=5
5260   GOSUB 6500:REM MOB SAVED.
5270   S9(11)=S9(11)+C1
5900   REM END OF SIMULATION.  ********************************************
5910   IF D9(C2)<C1 THEN 5999
5920   REM LPRINTER:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
5930   GOSUB 8800:REM DEBUG PRIN*.  **************************************
5940   REM CONSOLE:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
5999   NEXT N8
6000   REM
6002   REM RECORD RESULTS.  ********************************************
6003   REM
6004   REM  COMPUTE MEAN TIMES.
6005   REM
6006   FOR I=C1 TO 10
```

```
6008 IF S9(I)<=0THEN 6012
6010 T9(I)=T9(I)/S9(I)
6012 NEXT I
6014     IF D9(4)<C1THEN 6024
6016 REM  LPRINTER :REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
6018     GOSUB 6100:REM REGULARPRINT-OUT.******************************
6020 REM  CONSOLE :REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
6024     IF D9(5)<C1THEN 6070
6026 REM
6028 REM  COMPRESSED PRINT-OUT.  ****************************
6030 REM
6032     PRINT TAB(7);"SEEN = ";S8(C1);"  R.B. DEPLOYED = ";S8(4);
6034     PRINT TAB(43);"NOT SEEN = ";S8(C2);"  R.B. FAILED = ";S8(3)
6035% ####
6036 PRINTUSING 6035,N3;
6038% ##.###
6040     FOR I=C1TO 10
6042 PRINTUSING 6038,S9(I);
6044     NEXT I:PRINT :PRINT TAB(6);
6046     FOR I=C1TO 10
6052 PRINTUSING 6038,T9(I);
6054     NEXT I:PRINT :PRINT S9(11);TAB(6);
6056     FOR I=C1TO 10
6058 PRINTUSING 6038,M9(I);
6060     NEXT I:PRINT :PRINT TAB(6);
6062     FOR I=C1TO 10
6064 PRINTUSING 6038,N9(I);
6066     NEXT I:PRINT :PRINT :PRINT
6070 REM  WRITE TO DISK. *************************************
6074 DATA SAVE DC #2,S9(),T9()
6076 DATA SAVE DC #2,M9(),N9(),S8(1),S8(2),S8(3),S8(4),S9(11)
6099     GOTO 140
6100 REM
6110 REM  REGULAR PRINT-OUT.  ****************************************
6120 REM
6125%#.##
6130     L1=3:L2=L1+5:L3=L1+29:L4=L1+19:L5=L1+38:L6=L4+28
6140     PRINT :PRINT :PRINT
6142     GOSUB 7800:REM PRINTHEADINGWITHS/N.
6144     PRINT :PRINT :PRINT TAB(L1);T1$
6146     PRINT :PRINT TAB(L2);"LENGTH = ";T1(C1);"FEET";
6148     PRINT TAB(L5);"SPEED = ";T1(C2);K$
6150     PRINT TAB(L2);"NAVIGATIONAL HEADING ERROR = ";T1(12);"DEGREE(S)"
6152     PRINT TAB(L2);P$;"SHIP WILL BE RESTRICTED FROM TURNING = ";
6154   PRINTUSING 6125,P2(5):REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
6156     PRINT TAB(L2);P$;"SHIP WILL RECOVER MOB DIRECTLY = ";
6157   PRINTUSING 6125,P2(10):REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
6158     PRINT :PRINT :PRINT TAB(L1);R$
6160     PRINT TAB(L2);"SPEED:  ";D1(C1);K$;" IN CALM WATER"
6162     PRINT TAB(L2+8);D1(C2);K$;" IN 8' ";S$;" SEA"
```

```
6164    PRINT TAB(L2);"FREEBOARD = ";D1(3);"FEET WITH ";
6166    PRINT D1(4);"% OF GUNWALE OPEN"
6168    PRINT TAB(L2);"HEIGHT OF LOOKOUT'S EYE = ";D1(5);"FEET"
6170    PRINT TAB(L2);"MEAN TIME TO PREPARE ";R$;"FOR LAUNCH = ";D1(6);"MINUTES"
6172    PRINT TAB(L2); D1(7);"FALL(S) HAVING ";
6174    PRINT  D1(8);"FT./MIN. DESCENT SPEED"
6176    PRINT TAB(L2);"MAXIMUM SHIP SPEED FOR SAFE ";R$;"LAUNCH = ";D1(9);K$
6178    PRINT TAB(L2);"RATED SEA STATE CAPABILITY = ";D1(10);"FOOT ";S$
6180    PRINT TAB(L2);"MEAN TIME TO PERFORM STANDARD MANEUVER = ";D1(11);"MINUT
6182    PRINT TAB(L2);P$;R$;"WILL BE DEPLOYED TO SEARCH WITH SHIP = ";
6184    PRINT P2(6):PRINT:PRINT
6186    PRINT TAB(L1);"RESULTS OF ";N4;"SIMULATIONS:"
6188    PRINT:PRINT
6190    N=(S9(5)+S9(10))/N4
6192%##.###
6194    PRINT TAB(L1);"OVERALL FRACTION OF";TAB(L3);"********"
6196    PRINT TAB(L1);"MOB'S RESCUED";TAB(L3);"*";
6198    PRINT USING 6192,N;:PRINT "*"
6210    PRINT TAB(L1);"BY ";R$;TAB(L3);"********"
6220    PRINT:PRINT
6222    PRINT TAB(L1);R$;"DEPLOYED ";S8(4);"TIMES"
6224    PRINT TAB(L1);R$;"LAUNCH FAILED ";S8(3);"TIMES DUE TO SEA STATE"
6226    PRINT:PRINT
6252    PRINT TAB(L4);"  MOB SEEN ";S8(C1);"TIMES";
6254    PRINT TAB(L6);":  MOB NOT SEEN ";S8(C2);"TIMES"
6260    PRINT TAB(L6);":":PRINT TAB(L6);":"
6262    PRINT TAB(L4);H$;":";H$
6264    PRINT TAB(L4);" TIMES";TAB(L6);": TIMES"
6266    PRINT TAB(L6);":"
6268    PRINT TAB(L1);"MOB ALIVE WHEN:";TAB(L6);":":PRINT TAB(L6);":"
6270    PRINT TAB(L1);"BRIDGE NOTIFIED";TAB(L4);
6272    J8=C1:GOSUB 6400:REM PRINT LINE OF NUMBERS.
6274    PRINT TAB(L1);"MOB'S POS. REACHED";TAB(L4);
6276    J8=C2:GOSUB 6400
6280    PRINT TAB(L1);"MOB FOUND";TAB(L4);
6282    J8=3:GOSUB 6400
6286    PRINT TAB(L1);"ABOARD RESCUE BOAT";TAB(L4);
6288    J8=4:GOSUB 6400
6290    PRINT TAB(L1);"MOB SAVED";TAB(L4);
6292    J8=5:GOSUB 6400
6294    FOR I=C1 TO 5
6296    PRINT:NEXT I
6299    RETURN
6400    REM
6410    REM FORMATTED OUTPUT SUBROUTINE FOR REGULAR PRINT-OUT.  ********
6420    REM
6430    REM PRINTS NO., MEAN, MIN., AND MAX. FOR SEEN AND NOT SEEN.
6440    REM
6450%####.##
6452%#####
6460    FOR J7=C0 TO C1
6470      PRINT USING 6452,S9(J8);:PRINT USING 6450,T9(J8);
6472    IF ABS(N9(J8))>=10000 THEN 6476
```

```
6474     PRINT USING 6450,N9(J8);M9(J8);
6475     GOTO 6478
6476     PRINT "  ***    ***  ";
6478     J8=J8+5
6480     IF J7>C0 THEN 6490
6482   PRINT ":";
6490   NEXT J7
6492   PRINT:PRINT TAB(L6);":"
6499   RETURN
6500   REM
6510   REM ACCUMULATE STATISTICS.  ***************************************
6520   REM
6530   IF J9>5 THEN 6560
6540   IF F(C1)>C1 THEN 6560
6550   J9=J9+5
6560   S9(J9)=S9(J9)+C1
6570   T9(J9)=T9(J9)+T
6580   IF T>N9(J9) THEN 6590
6585   N9(J9)=T
6590   IF T<M9(J9) THEN 6599
6595   M9(J9)=T
6599   RETURN
7000   REM SET UP T3 ARRAY ELEMENTS.  *************************
7010   REM
7030   REM  Z=1 FOR SEARCH WITH SHIP WHEN MOB 'SEEN'.
7040   REM  Z=2 FOR SEARCH WITH BOAT WHEN MOB 'SEEN'.
7050   REM  Z=3 FOR SEARCH WITH SHIP WHEN MOB 'NOT SEEN'.
7060   REM  Z=4 FOR SEARCH WITH BOAT WHEN MOB 'NOT SEEN'.
7070   REM
7080   REM SET UP T3(1) & T3(2) DEPENDING UPON Z.
7085   REM
7090   ON Z GOTO 7091,7092,7093,7094
7091   GOSUB 7200:GOTO 7095
7092   GOSUB 7300:GOTO 7095
7093   GOSUB 7400:GOTO 7095
7094   GOSUB 7300
7095   ON Z GOTO  7100,7100,7500,7500
7100   REM
7110   REM SET UP REMAINDER OF T3 ARRAY FOR 'SEEN'.
7115   REM
7120   T3(3)=T5(10)-T
7130   T3(4)=N2
7140   T3(5)=C0
7150   T3(6)=0.5
7160   T3(7)=C0
7170   T3(8)=0.5*RND(1)
7180   T3(9)=INT(T3(3)-T3(C1)-T3(C2))
7182   IF Z<C2 THEN 7190
7184   T3(9)=INT(T3(9)/V(4))
7190   T3(10)=P2(12)
7199   RETURN
7200   REM
7210   REM SET UP T3(1) & T3(2) FOR SHIP--SEEN.
```

```
7220    REM
7230    T3(C1)=T5(13)
7240    T3(C2)=T5(5)+0.5*T5(8)
7299    RETURN
7300    REM
7310    REM SET UP TE(1) & T3(2) FOR SEARCH WITH BOAT.
7322    REM
7324    REM TIME TO REPORT TO R.B.
7326    REM
7330    R(C1)=C1:GOSUB 8100:T5(6)=R3
7332    REM
7334    REM TIME TO PREPARE R.B.
7336    REM
7340    R(C1)=D1(6):GOSUB 8100:T5(7)=R3
7350    R3=R3+T5(6)
7352    IF R3>T5(8) THEN 7356
7354    R3=T5(8)
7356    T3(C1)=R3
7360    REM
7362    REM TIME TO LAUNCH R.B.
7364    REM
7366    T5(14)=((T1(8)+15)/D1(8))+0.5
7370    IF Z>C2 THEN 7390
7380    T3(C2)=T5(14)+(T5(5)+0.5*T5(8))*V(4)
7389    RETURN
7390    T3(C2)=T5(14)+(T5(C2)+T5(5)+0.5*T5(8))*V(4)
7399    RETURN
7400    REM
7410    REM SET UP T3(1) & T3(2) FOR SHIP--NOT SEEN.
7420    REM
7430    T3(C1)=T5(9)
7440    T3(C2)=T5(C2)+T5(5)
7499    RETURN
7500    REM
7510    REM SET UP REMAINDER OF T3 ARRAY FOR 'NOT SEEN'.
7512    REM
7514    REM Z=3 FOR SHIP; Z=4 FOR BOAT.
7516    REM
7520    T3(3)=T5(10)-T
7522    T3(4)=N2
7530    T3(6)=T5(C1)
7535    T3(7)=T5(13)
7540    T3(8)=T5(C2)
7545    T3(9)=INT((T3(3)-T3(C1)-T3(C2))/(T5(C1)+T5(13)))
7550    T3(10)=P2(8)
7555    REM GET DISTANCE OF VISIBILITY & COMPLETE T3(9).
7560    REM Z=3 FOR SHIP; Z=4 FOR BOAT.
7565    ON Z-C2 GOTO 7570,7580
7570    T3(5)=V(5)
7574    REM COMPLETE T3(9).
7576    T3(9)=T3(9)-C1
7579    RETURN
7580    T3(5)=V(6)
```

```
7585     REM COMPLETE T3(9).
7590     T3(9)=INT(T3(9)/V(4))-C1
7599     RETURN
7600     REM DEBUG #3 PRINT-OUT. ****************************
7610     PRINT TAB(30);"MAN OVERBOARD SIMULATION":PRINT
7620     PRINT TAB(49);"SIMULATION SERIAL NO. MOB";N3:PRINT:PRINT
7640     PRINT "SHIP: ";N1;"  ";T1(1);"  ";T1(C2);"  ";T1(5);
7650     PRINT "  ";T1(12);"  ";T1$;"  ";N4;" SIMULATIONS."
7660     PRINT "R.B.:";
7670% ###.#
7675     W=6:D=1
7680     FOR J=C1 TO 12
7684     PRINT USING 7670,D1(J);:REM WWWWWWWWWWWWWWWWWWWWWWWWW
7688     NEXT J
7690     PRINT:PRINT
7699     RETURN
7800     REM
7810     REM HEADING FOR REGULAR PRINT-OUT.  *****************************
7820     REM
7830     PRINT TAB(30);"MAN OVERBOARD SIMULTION":PRINT
7840     PRINT TAB(49);"SIMULATION SERIAL NO. MOB";N3:PRINT:PRINT
7899     RETURN
7900     REM
7910     REM HEADINGS FOR COMPRESSED PRINT-OUT.  ***************************
7920     REM
7925     PRINT:PRINT
7930     PRINT " S/N  NOTIFY REACH P FIND  IN R.B. SAVED";
7940     PRINT "  NOTIFY REACH P FIND  IN R.B. SAVED"
7950     PRINT:PRINT
7999     RETURN
8000     REM NORMAL DISTRIBUTION.  *******************************
8010     R1=RND(1)
8020     IF R1>C0 THEN 8030
8025     R1=L9
8030     R2=RND(1)
8040     V1=SQR(-C2*LOG(R1))*COS(C9*R2)
8050     R3=V1*R(4)+R(C1)
8060     IF R3>R(C2) THEN 8070
8065     R3=R(C2)
8070     IF R3<R(3) THEN 8099
8080     R3=R(3)
8099     RETURN
8100     REM FRANK NICKELS' NORMAL DIST.  ***********************
8110     R(2)=C0
8120     R(3)=C2*R(C1)
8130     R(4)=0.5*R(C1)
8199     GOTO 8000:REM COMPUTE NORMAL DIST. & RETURN.
8200     REM RAYLEIGH DIST.  ************************************
8210     R1=RND(1)
8220     IF R1>C0 THEN 8230
8225     R1=L9
8230     R3=R(C1)*SQR(-C2*LOG(R1))
8299     RETURN
```

```
8300    REM UNIFORM DIST.  ****************************************
8310    R3=R(C2)+(R(3)-R(2))*RND(1)
8399    RETURN
8400    REM SURVIVAL TIME (HYPERTHERMIA).  *********************
8410    R(C2)=C8*(V(C1)*(S1+S2*V(C1)*V(C1))-S3)
8420    R(3)=C8*(V(C1)*(S4*V(C1)*V(C1)-S5)+S6)
8430    R(C1)=0.5*(R(C2)+R(3))
8440    R(4)=0.25*(R(3)-R(C2))
8499    GOTO 8000:REM COMPUTE NORMAL DIST. & RETURN.
8500    REM SURVIVAL TIME (FLOTATION).  ************************
8510    IF RND(1)<=P THEN 8560
8520    REM MAN NOT WEARING PFD.  --------------------------------
8530    F(C2)=C2
8540    R(1)=T5(3)
8550    GOTO 8200:REM COMPUTE RAYLEIGH DIST. & RETURN.
8560    REM MAN WEARING PFD.  ------------------------------------
8570    F(C2)=C1
8580    R(C1)=T5(4)
8599    GOTO 8200:REM COMPUTE RAYLEIGH DIST. & RETURN.
8700    REM DEBUG PRINTOUTS.  **************************************************
8710    PRINT "CONSTANT DURING A GIVEN SIMULATION."
8720    PRINT "SEASON";TAB(17);"WATER TEMP.";
8730    PRINT TAB(33);"WAVE HEIGHT";TAB(49);"T(S) H.T.";
8740    PRINT TAB(65);"A FOR N.E."
8745    CONVERT S TO S$,(#############)
8750    PRINT S$;TAB(17);V(1);TAB(33);V(2);
8760    PRINT TAB(49);T5(10);TAB(65);N2
8770    PRINT:PRINT
8799    RETURN
8800    REM DEBUG PRINT-OUT (END  OF EACH SIMULATION). *******************
8820%###
8830%#####.###
8835    D=C0:W=3
8840    FOR J=C1 TO 6
8850    PRINT USING 8820,F(J);:REM  WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
8860    NEXT J
8880    PRINT
8890    PRINT USING 8830,T;T0;R3;V(C2):REM WWWWWWWWWWWWWWWWWWWWWWWW
8899    RETURN
9000    REM INITIALIZE  ****************************************
9005    B9=1E30
9010    C0=0:C1=1:C2=2
9014    C3=11:C4=10
9016    C8=60:C9=3.14159
9020    C1$=" CHARACTERISTICS."
9022    D0(C1)=1:D0(C2)=10:D0(3)=100:D0(4)=1000:D0(5)=10000
9024    D0(6)=1.E5:D0(7)=1.E6:D0(8)=1.E7:D0(9)=1.E8:D0(10)=1.E9
9030    F1=254:G1=6
9032    H$="  NO.    MEAN   MIN.   MAX. "
9034    K$="KNOTS"
9036    L9=1E-30
9040    M1=33:M2=12
9045    M3=13:M4=17:M5=8
```

```
9050    P3=0.9595:P4=0.0095
9052    P5=0.0168:P6=0.0008
9060    P$="PROBABILITY "
9062    R$="RESCUE BOAT "
9064    S1=0.0024:S2=8.1E-6:S3=0.1
9066    S4=5.9E-5:S5=0.101:S6=2.25
9070    S$="SIGNIFICANT WAVE HEIGHT"
9080    X0=19
9090    REM SET UP BLANK VARIABLE.
9092    B$="                         "
9099    RETURN
9200    REM HOME & ERASE TV.  *********************************************
9210    PRINT CHR$(5)
9220    RETURN
9400    REM
9410    REM CORE OF SIMULATION.  ********************************
9412    REM
9414    REM  1.  SHIP SLOWS OR TURNS.
9416    REM  2.  SHIP OR BOAT RETRACES SHIP'S PATH.
9418    REM  3.  SHIP OR/AND BOAT SEARCHES FOR MOB.
9420    REM  4.  SHIP OR BOAT RECOVERS MOB OR MOB LOST.
9422    REM
9430    T2=T3(C1)+T3(C2):T4=T2
9432    IF T2>T3(3) THEN 9465
9434    IF T3(4)<=T3(5) THEN 9480
9436    IF T3(9)<C1 THEN 9475
9440    FOR K=C1 TO T3(9) STEP C2
9442      T2=T2+(C2*(T3(6)-T3(8))+T3(7))/V(4)
9444      IF RND(1)<=T3(10) THEN 9485
9446      IF K=T3(9) THEN 9470
9448      T2=T2+(C2*T3(8)+T3(7))/V(4)
9450      IF RND(1)<=T3(10) THEN 9490
9452    NEXT K
9460    F(4)=3:RETURN
9465    F(4)=1:RETURN
9470    F(4)=2:RETURN
9475    F(4)=5:RETURN
9480    F(4)=8:GOTO 9499
9485    F(4)=9:GOTO 9499
9490    F(4)=10
9499    RETURN
9500    REM DISTANCE OF VISIBILITY FROM SHIP.  *****************
9510    V(5)=1000+100*T1(4)
9512    IF V(5)<2500 THEN 9520
9514    V(5)=2500
9520    IF T5(15)>V(5) THEN 9550
9530    V(5)=T5(15)
9550    REM DISTANCE OF VISIBILITY FROM BOAT.  *****************
9560    V(6)=1000+100*D1(5)
9562    IF V(6)<2500 THEN 9570
9564    V(6)=2500
9570    IF T5(15)>V(6) THEN 9599
9580    V(6)=T5(15)
9599    RETURN
```

APPENDIX F: Man Overboard Output Program Listing

```
1       REM MAN OVERBOARD REGULAR PRINT-OUT PROGRAM       2 MARCH 1978
10      REM
12      REM THIS PROGRAM SERVES TWO PURPOSES.
14      REM
16      REM  1.  TO PROVIDE A STAND-ALONE PROGRAM TO PRODUCE THE REGULAR
18      REM         PRINT-OUT REPORTS FROM THE OUTPUT SAVED ON DISK.
20      REM
22      REM  2.  TO PROVIDE A SUBSTITUTE FOR THE REGULAR PRINT-OUT
24      REM         IN THE MAN OVERBOARD SIMULATION PROGRAM.
26      REM
28      REM
30      REM  THE SUBROUTINES 6100-6299 AND 6400-6499 CAN REPLACE THE
32      REM     CORRESPONDING SUBROUTINES IN THE MAN OVERBOARD
34      REM     SIMULATION PROGRAM.
36      REM     THE SUBROUTINE 7700-7799 IN THE MAN OVERBOARD SIMULATION
38      REM     PROGRAM CAN THEN BE DELEETED.
40      REM
60      DIM B$64,H$64,K$64,P$64,R$64,S$64
70      SELECT PRINT 005(80):PRINT HEX(03)
100     DIM D0(10),D1(15),M9(10),N9(10),P2(15)
110     DIM S8(4),S9(11),T1(12),T5(15),T9(10)
120     C0=0:C1=1:C2=2
122     D0(C1)=C1:D0(C2)=10:D0(3)=100:D0(4)=1000
124     D0(5)=1E4:D0(6)=1E5:D0(7)=1E7
126     D0(8)=1E7:D0(9)=1E8:D0(10)=1E9
130     INPUT "SIMULATION OUTPUT FILE NAME";A$
140     SELECT #1 B20
150     DATA LOAD DC OPEN R#1,A$
152     B$="                                    "
154     H$="  NO.     MEAN    MIN.    MAX. "
155     K$="KNOTS"
156     P$="PROBABILITY "
158     R$="RESCUE BOAT "
159     S$="SIGNIFICANT WAVE HEIGHT"
178     DATA LOAD DC #1,N1,N2,N3,N4:IF END THEN 500:REM WWWWWWWWWWWWWWW
180     FOR I=C1 TO 12
190       DATA LOAD DC #1,T1(I),D1(I),P2(I),T5(I):REM WWWWWWWWWWWWWWWWWW
200     NEXT I
210     FOR I=13 TO 15
220       DATA LOAD DC #1,D1(I),P2(I),T5(I):REM WWWWWWWWWWWWWWWWWWWWWWWW
230     NEXT I
240     DATA LOAD DC #1,T1$
250     FOR I=C1 TO 10
260       DATA LOAD DC #1,S9(I),T9(I),M9(I),N9(I):REM  WWWWWWWWWWWWWWWWW
270     NEXT I
280     DATA LOAD DC #1,S8(1),S8(2),S8(3),S8(4):REM WWWWWWWWWWWWWWWWWWWW
290     DATA LOAD DC #1,S9(11):REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
300     SELECT PRINT 215(160)
310     GOSUB 6100:REM PRINT OUTPUT REPORT.
320     SELECT PRINT 005(80)
330     GOTO 170
400     PRINT "CAN'T OPEN FILE."
410     STOP
```

```
500     PRINT "FINISHED."
510     DATA SAVE DC CLOSEALL
520     STOP
600     PRINT "DUMB FILE ERROR."
610     STOP
6100    REM
6110    REM REGULAR PRINT-OUT.  ****************************************
6120    REM
6125%#.##
6130    L1=3:L2=L1+5:L3=L1+29:L4=L1+19:L5=L1+38:L6=L4+28
6140    PRINT:PRINT:PRINT
6142    GOSUB 7800:REM PRINT HEADING WITH S/N.
6144    PRINT:PRINT:PRINT TAB(L1);T1$
6146    PRINT:PRINT TAB(L2);"LENGTH = ";T1(C1);"FEET";
6148    PRINT TAB(L5);"SPEED = ";T1(C2);K$
6150    PRINT TAB(L2);"NAVIGATIONAL HEADING ERROR = ";T1(12);"DEGREE(S)"
6152    PRINT TAB(L2);P$;"SHIP WILL BE RESTRICTED FROM TURNING = ";
6154    PRINT USING 6125,P2(5):REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
6156    PRINT TAB(L2);P$;"SHIP WILL RECOVER MOB DIRECTLY = ";
6157    PRINT USING 6125,P2(10):REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
6158    PRINT:PRINT:PRINT TAB(L1);R$
6160    PRINT TAB(L2);"SPEED:  ";D1(C1);K$;" IN CALM WATER"
6162    PRINT TAB(L2+8); D1(C2);K$;" IN 8' ";S$;" SEA"
6164    PRINT TAB(L2);"FREEBOARD = ";D1(3);"FEET WITH ";
6166    PRINT D1(4);"% OF GUNWALE OPEN"
6168    PRINT TAB(L2);"HEIGHT OF LOOKOUT'S EYE = ";D1(5);"FEET"
6170    PRINT TAB(L2);"MEAN TIME TO PREPARE ";R$;"FOR LAUNCH = ";D1(6);"MINUTES"
6172    PRINT TAB(L2); D1(7);"FALL(S) HAVING ";
6174    PRINT  D1(8);"FT./MIN. DESCENT SPEED"
6176    PRINT TAB(L2);"MAXIMUM SHIP SPEED FOR SAFE ";R$;"LAUNCH = ";D1(9);K$
6178    PRINT TAB(L2);"RATED SEA STATE CAPABILITY = ";D1(10);"FOOT ";S$
6180    PRINT TAB(L2);"MEAN TIME TO PERFORM STANDARD MANEUVER = ";D1(11);"MINUTE
6182    PRINT TAB(L2);P$;R$;"WILL BE DEPLOYED TO SEARCH WITH SHIP = ";
6184    PRINT P2(6):PRINT:PRINT
6186    PRINT TAB(L1);"RESULTS OF ";N4;"SIMULATIONS:"
6188    PRINT:PRINT
6190    N=(S9(5)+S9(10))/N4
6192%##.###
6194    PRINT TAB(L1);"OVERALL FRACTION OF";TAB(L3);"********"
6196    PRINT TAB(L1);"MOB'S RESCUED";TAB(L3);"*";
6198    PRINT USING 6192,N;:PRINT "*":REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
6210    PRINT TAB(L1);"BY ";R$;TAB(L3);"********"
6220    PRINT:PRINT
6222    PRINT TAB(L1);R$;"DEPLOYED ";S8(4);"TIMES"
6224    PRINT TAB(L1);R$;"LAUNCH FAILED ";S8(3);"TIMES DUE TO SEA STATE"
6226    PRINT:PRINT
6252    PRINT TAB(L4);"  MOB SEEN ";S8(C1);"TIMES";
6254    PRINT TAB(L6);":  MOB NOT SEEN ";S8(C2);"TIMES"
6260    PRINT TAB(L6);":":PRINT TAB(L6);":"
6262    PRINT TAB(L4);H$;":";H$
6264    PRINT TAB(L4);" TIMES";TAB(L6);": TIMES"
6266    PRINT TAB(L6);":"
6268    PRINT TAB(L1);"MOB ALIVE WHEN:";TAB(L6);":":PRINT TAB(L6);":"
```

```
6270     PRINT TAB(L1);"BRIDGE NOTIFIED";TAB(L4);
6272     J8=C1:GOSUB 6400:REM PRINT LINE OF NUMBERS.
6274     PRINT TAB(L1);"MOB'S POS. REACHED";TAB(L4);
6276     J8=C2:GOSUB 6400
6280     PRINT TAB(L1);"MOB FOUND";TAB(L4);
6282     J8=3:GOSUB 6400
6286     PRINT TAB(L1);"ABOARD RESCUE BOAT";TAB(L4);
6288     J8=4:GOSUB 6400
6290     PRINT TAB(L1);"MOB SAVED";TAB(L4);
6292     J8=5:GOSUB 6400
6294     FOR I=C1 TO 5
6296     PRINT:NEXT I
6299     RETURN
6400     REM
6410     REM FORMATTED OUTPUT SUBROUTINE FOR REGULAR PRINT-OUT.  ********
6420     REM
6430     REM PRINTS NO., MEAN, MIN., AND MAX. FOR SEEN AND NOT SEEN.
6440     REM
6450%####.##
6452%#####
6460     FOR J7=C0 TO C1
6470       PRINT USING 6452,S9(J8);:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
6471       PRINT USING 6450,T9(J8);:REM WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
6472       IF ABS(N9(J8))>=10000 THEN 6476
6474       PRINT USING 6450,N9(J8);M9(J8);:REM WWWWWWWWWWWWWWWWWWWWWWWWW
6475       GOTO 6478
6476       PRINT "  ***     ***  ";
6478       J8=J8+5
6480       IF J7>C0 THEN 6490
6482     PRINT ":";
6490     NEXT J7
6492     PRINT:PRINT TAB(L6);":"
6499     RETURN
7800     REM
7810     REM HEADING FOR REGULAR PRINT-OUT.  ****************************
7820     REM
7830     PRINT TAB(30);"MAN OVERBOARD SIMULATION":PRINT
7840     PRINT TAB(49);"SIMULATION SERIAL NO. MOB";N3:PRINT:PRINT
7899     RETURN
```