

AD-A127 424

AN INTERACTIVE COMPUTER CODE FOR WATER WAVES IN OPEN
CHANNELS(U) NAVAL RESEARCH LAB WASHINGTON DC
N C CHU ET AL. 13 APR 83 NRL-MR-5066

1/1

UNCLASSIFIED

F/G 9/2

NL

END

FILMED

101

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

NRL Memorandum Report 5066

AD A127424

An Interactive Computer Code for Water Waves in Open Channels

A. G. CHU AND G. A. KERAMIDAS

Fluid Dynamics Branch
Naval Research Laboratory

April 13, 1983



NAVAL RESEARCH LABORATORY
Washington, D.C.

DTIC
ELECTE
APR 29 1983
S D E

88 04 29 045

Approved for public release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 5066	2. GOVT ACCESSION NO. AD-A127 424	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN INTERACTIVE COMPUTER CODE FOR WATER WAVES IN OPEN CHANNELS		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing problem
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) N. C. Chu and G. A. Keramidas		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N23, RR0230141 1470-00
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217		12. REPORT DATE April 13, 1983
		13. NUMBER OF PAGES 32
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Finite elements Computer codes Water waves Shallow water equations		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A computational model has been developed in the Fluid Dynamics Branch of the Naval Research Laboratory for the simulation of water waves in open channels. The model numerically solves a set of partial differential equations known as the Shallow Water equations. The method employed for the solution of these equations is the Finite Element Method with linear element approximations in time and space. The computer code was initially developed for the computational model on a HP-1000 computer system. This report contains the interactive version of the computer code, its description and provides information on how the code can be used.		

DD FORM 1473
1 JAN 73

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

I. INTRODUCTION	1
II. COMPUTATIONAL MODEL	1
1. Analytical Model	1
2. Computer Model	2
III. COMPUTER PROGRAM	2
3. Program Description	2
4. Program Verification	4
IV. CONCLUDING REMARKS	5
ACKNOWLEDGMENT	5
REFERENCES	5

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	



COMPUTATIONAL HYDRODYNAMICS: AN INTERACTIVE COMPUTER CODE FOR WATER WAVES IN OPEN CHANNELS

I. INTRODUCTION

The analysis and solution of free surface wave problems requires the development of a model which can represent the physical phenomenon under study. However the complexity of the equations describing surface waves imposes certain restrictions in the type of solution that one can use. Most often these equations are solved by some approximate method which requires the development of a computational model implementing such a method.

For the purpose of studying water waves in open channels a computational model has been developed. The analytical formulation and the description of the model can be found in [1]. The one-dimensional form of the shallow water equations is solved numerically by the finite element method (FEM). The simplest form of approximation in space and time is utilized for the FEM and the derived model is capable of producing accurate solutions to a variety of problems with free surface motion [2,3].

The numerical solution of the system of equations which represent the computational model is obtained by a computer code developed on an HP 1000 computer. A modification of the code was necessary in order to utilize the code in an interactive way and to be able to solve different types of cases with the least amount of user effort. Following a brief description of the analytical model in the first part of this report, the computer model is presented in the second part. The different parts of the computer model are given in a flow chart diagram and for each part of the required subroutines are described. In the last part of the report the solutions of some example problems are given. A user's manual and a listing of the computer code are included in Appendices A and B respectively.

II. COMPUTATIONAL MODEL

1. Analytical Model

The motion of a body of water with a free surface in a channel can be described by a set of two nonlinear partial differential equations (shallow water equations). The one dimensional form of these equations is

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x} (VH) = 0 \quad (1)$$

$$\frac{\partial V}{\partial t} + V \frac{\partial V}{\partial x} + g \frac{\partial h}{\partial x} = 0 \quad (2)$$

where h is the instantaneous water surface elevation from the mean surface level, H is the total depth of the fluid, V is the depth-averaged velocity in the x -direction, t is the time and g is the gravitational acceleration. Equations (1) and (2) are cast into a variational form and the FEM is applied to derive a system of ordinary differential equations given by

$$[M]\{\dot{q}\} + [K]\{q\} = \{Q\} \quad (3)$$

where $[M]$ is the mass matrix, $[K]$ is the convective matrix, $\{Q\}$ is a vector representing boundary forces and $\{q\}$ is the vector representing either $\{h\}$ or $\{V\}$.

The system of equations represented by Eq. (3) is reduced to an algebraic system by approximating the time derivative and the system then is given by

$$[A]\{q^{n+1}\} = \{Q\} + [B]\{q^n\} \quad (4)$$

where the matrices $[A]$ and $[B]$ are combinations of the matrices $[M]$ and $[K]$. The vectors $\{q^{n+1}\}$ and $\{q^n\}$ represent the unknowns at time levels $(n + 1)$ and (n) respectively. The derivation of the above equations can be found in [1].

The system of equations represented by Eq. (4) is solved numerically by a computer code developed at NRL for that purpose. The different parts of the code are described in the following section.

2. Computer Model

The analytical model described in the previous section can be used to solve a variety of physical problems concerned with water waves. Solutions are obtained by casting the analytical model into a computer model. A general description of the computer model is given in the flow chart, Fig. 1. The different stages of the model generate the required information and perform the numerical solution for the specified problem.

In the first stage the physical problem under consideration is defined. This includes the particular channel geometry, boundary conditions and initialization of all program parameters. The subroutines called in this stage are the INPUT and GEOMT subroutines. In the next stage the system of algebraic equations is formed (subroutine ELMAT) and the system is solved (subroutine QSOLV) for the particular type of boundary conditions under consideration (subroutines MPART, DCOMP, and SOLVE). Following the solution, the values for the surface elevation, velocity and pressure are computed. The process is repeated for every time increment. The last stage of the code performs such tasks as output of results, plotting of results either at specified time intervals or at the end of the solution. This stage includes subroutines XOUT, XPLOT, and TOUT, TPLOT.

As was mentioned earlier the noninteractive version of the computer code can be used on any computer. The interactive version of the code utilizes the HP-1000 graphics terminal 2648 and certain changes will be necessary in order for the code to be used on another machine.

The code has been optimized for minimum storage requirements and execution time. Further optimization can be achieved by utilizing the vector instruction set (VIS) of the HP-1000 computer system. Although this reduces the execution time, it also restricts the transferability of the code to other machines.

The description of each of the subroutines is given in the next section.

III. COMPUTER PROGRAM

3. Program Description

The computer program is composed by the main program (program KYMA) and eleven subroutines. The dimensions of the arrays are set for a maximum of 222 equations. This allows for 111 nodal points and two unknowns per point. The storage requirements for the set dimensions is about 26 K words or 26 pages in the HP-1000 system. If a larger number of nodal points is required, then the

extended memory (EMA) of the system has to be used for additional storage. The operating system of the HP-1000 computer on which the code is running is the RTE-IVB.

Program KYMA

This is the main part of the code and is used as a driver for the calling sequence to the different subroutines. The definitions of the names of the variables and parameters used in the code are given in the listing of the code in the appendix. The first parameter that needs to be defined is the logical unit (LU) number for the input device, i.e., $LU = 1$, the input device is the users terminal. The call to INPUT sets the parameters and defines the problem to be solved. Next the user has to specify the time for starting the output of the results (TSTART) and the time to stop this task (TSTOP). If $TSTART \leq TIME \leq TSTOP$ then subroutines XOUT is called. For each time increment a call to QSOLV is required for the solution of the system of equations. This process is continued until the solution reaches the specified maximum time for the simulation.

INPUT

This is the primary parameter definition subroutine. The number of elements for the space discretization, the length of the channel and the water depth are first defined. Next the time increment Δt is defined together with the maximum time for the solution. The call to GEOMT will define the particular channel geometry. Following that step the user has to specify the type of boundary condition to be imposed at the inlet of the channel. At the outlet of the channel the boundary condition is adjusted accordingly by the code.

NOTE: The number of elements should be an even number. In the case of the half cylinder geometry this number should be a multiple of twenty.

GEOMT

This subroutine is used to generate the channel geometry. First the user specifies the type of geometry desired (more types can be added), then whether the channel is of infinite length (open channel) or finite length (closed channel) is considered. After the generation of the specified geometry a warning is given if the number of elements has been increased. This increase is necessary if the spacing of the nodes has been changed to accommodate a particular geometry. The different types of geometries currently generated by the code are shown in Fig. 2.

ELMAT

This subroutine is called by QSOLV and is used to define the element matrices $[M]$ and $[K]$, Eq. (3), and to assemble the element matrices $[A]$ and $[B]$.

QSOLV

This subroutine performs a number of tasks. First the boundary conditions for the specified type are defined. Then the boundary force vector is computed and the global matrices are assembled from the element matrices defined in ELMAT. A sequence of calls to MPART, DCOMP, and SOLVE will partition the global matrices according to the boundary conditions and will solve the system of equations, respectively.

XOUT

This subroutine is called by the main program and it is used for the output of results at selected devices. The user has the choice of output on a line printer, on magnetic tape or at the user's terminal.

XPLOT

This subroutine is used to plot the results at selected time intervals and it is called by XOUT. Results for the elevation, the velocity or the pressure can be plotted individually or in any combination.

TOUT

This subroutine is also called by the main program and it is used for the output of results as a function of time and at specific points with respect to the x-coordinate. Currently the values of the elevation and velocity at $x = 0$ and $x = L = (100.00)$ are obtained.

TPLOT

This subroutine is called by TOUT for plotting results at the user's terminal. Both XPLOT and TPLOT are written by utilizing the capabilities of the HP-2648 graphics terminal.

MPAPT

This subroutine performs the partitioning of the matrix coefficient of the system of equations according to the specified boundary conditions.

DCOMP and SOLVE

These two subroutines perform the decomposition and solution of the matrix systems of equations.

The last three subroutines are part of a users library on the HP-1000 computer system. This reduces the size of the computer code and the compiling time.

4. Program Verification

The required parameters for solving a test case with the developed computer code are specified by the user through subroutine INPUT.

The example which will be used as a demonstration case simulates the propagation of a surface wave in a channel with a half-cylinder placed on the channel bottom. A list of all the input parameters as they appear on the users terminal is given in Appendix A.

The first parameter is the number of elements (NUMEL) to be used for the space discretization, NUMEL = 100. The length of the computational domain is XL = 100.00 m and the water depth is $H_0 = 10.0$ m. The length of each element is equal to 1 m and the time step size is specified as $Dr = 0.02$. The channel geometry is given in Fig. 3 for the problem under consideration. For this type of geometry NUMEL has been increased to 110. This is a result of the finer mesh where the half-cylinder is positioned. The element length has been reduced to 0.5 at that location.

The physical problem is defined by specifying the type of boundary conditions at $x = 0.0$. In the present case the velocity is specified as

$$V(0, t) = V_0 \sin(\omega t)$$

where the amplitude $V_0 = 0.1$ and $\omega = 2\pi/T$. The wave period T is equal to 5.0 which corresponds to a wavelength of 50 or five times the depth H_0 .

The rest of the program parameters are needed for the output of results on a printer and/or tape and for the plotting results on the users terminal. Examples of the plotted results are given in Figs. 4 through 6. In Fig. 4 a continuous plot is given for the surface elevation. On the left side of the cylinder one can observe the formation of the standing wave due to the wave reflections from the cylinder. The actual surface elevation as well as the elevation with respect to the size of the channel are given on the same figure. The next two figures are for the elevation and velocity at $x = 0.0$ and $x = 100.0$. The change in elevation due to wave reflections can be seen in Fig. 5. These results have been compared to analytical and experimental ones and the agreement was very satisfactory [2].

The particular problem presented here is only an example of the type of problems that can be solved with the developed code. Additional types of geometries and boundary conditions can be easily handled by the code.

IV. CONCLUDING REMARKS

A finite element model has been implemented for simulating the motion of surface waves in open channels. The computer code which was developed for that purpose has been presented in this report. The description of the code together with an example problem provide the prospective user with sufficient information for implementing and using the code. At this point the code can be used on any machine with a FORTRAN compiler. The only exception is for the two plotting subroutines. These are designed to be used only with the HP-2648 terminals. The code has been optimized to a certain degree but further savings in execution time can be achieved by implementing the vector instruction set (VIS) available on the HP-1000 computer system.

ACKNOWLEDGMENT

The authors acknowledge the support of the Naval Research Laboratory and the Summer Research Apprentice Program for the work reported in this report.

REFERENCES

1. G.A. Keramidas, "A Hydrodynamic Model for Open Channel Flow Problems," Finite Elements in Water Resources, 4th Int'l. Conf., Hannover, Germany, (1982).
2. G.A. Keramidas and S.E. Ramberg, "Numerical Experiments with Reflecting Water Waves," in *Computational Methods and Experimental Measurements*, Washington, D.C., (1982).
3. G.A. Keramidas, "Computational Hydrodynamics: A Model for the Shallow Water Equations," NRL Memorandum Report.

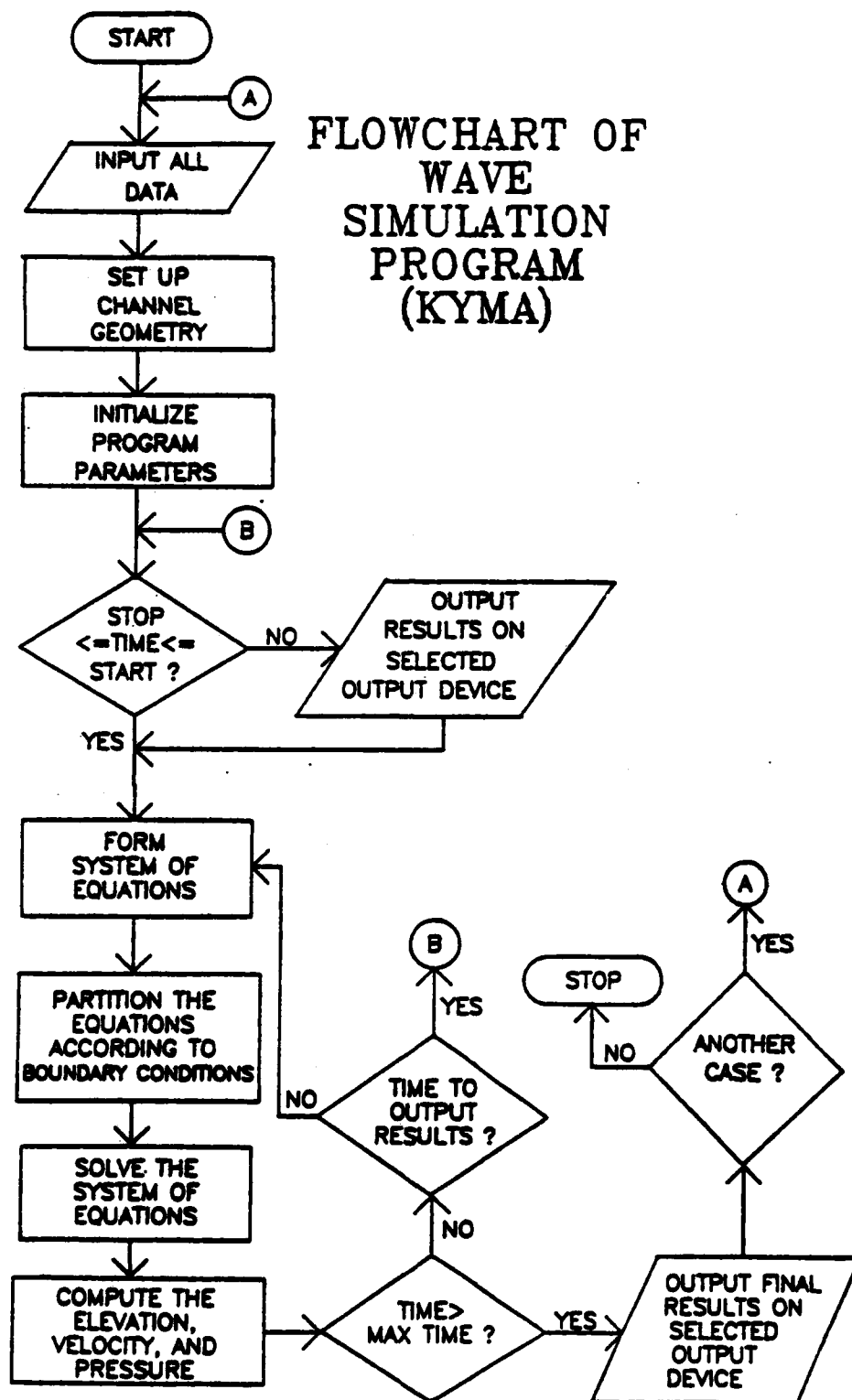
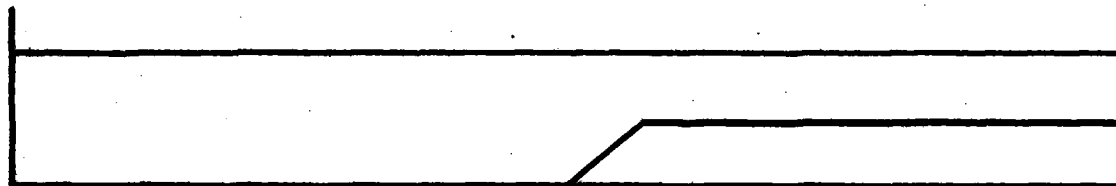


Fig. 1 — Flowchart of computer code

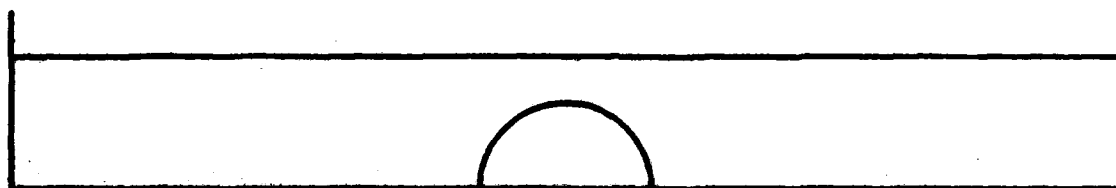
CHANNEL GEOMETRIES



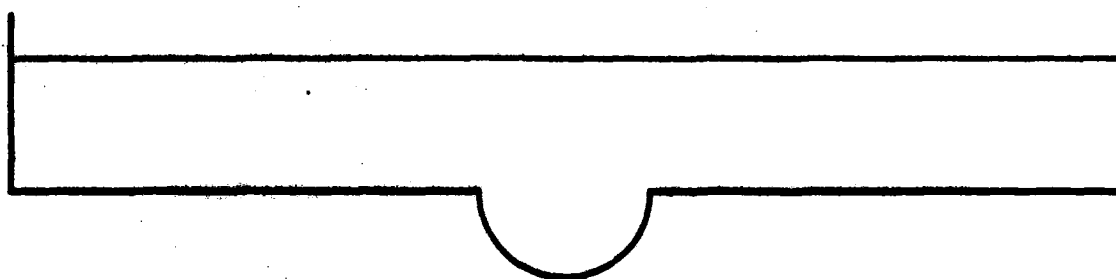
CHANNEL WITH A CONSTANT BOTTOM SLOPE



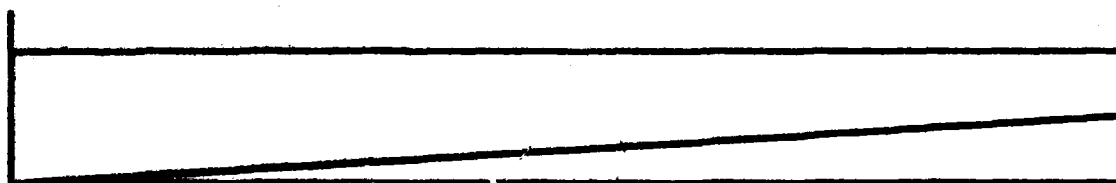
CHANNEL WITH A RAMP



CHANNEL WITH A HALF CYLINDER



CHANNEL WITH A TRENCH



CHANNEL WITH THE BOTTOM SLOPED UPWARD

Fig. 2 - Channel geometries presently available in the computer code

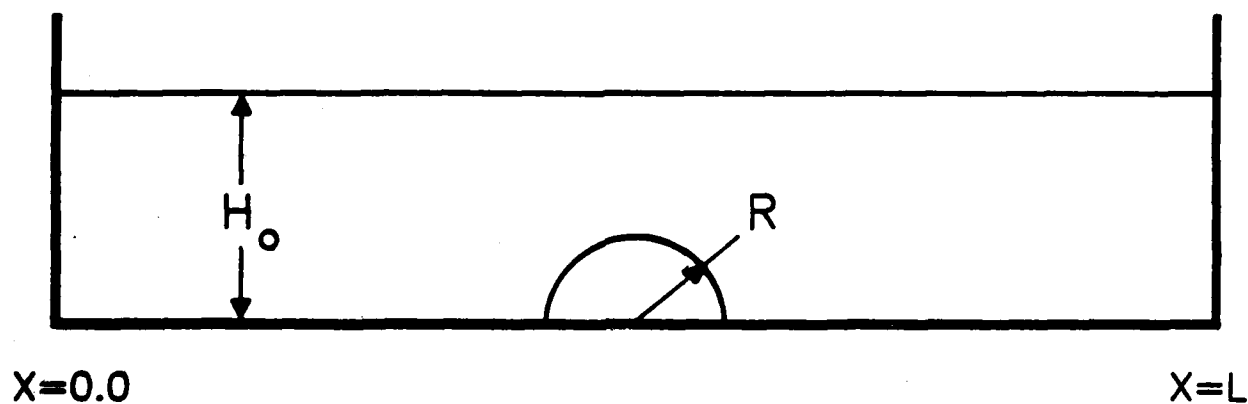


Fig. 3 — Channel geometry for example problem. Half-cylinder with radius $R = 5.0$ m and $H_0 = 10.0$ m.

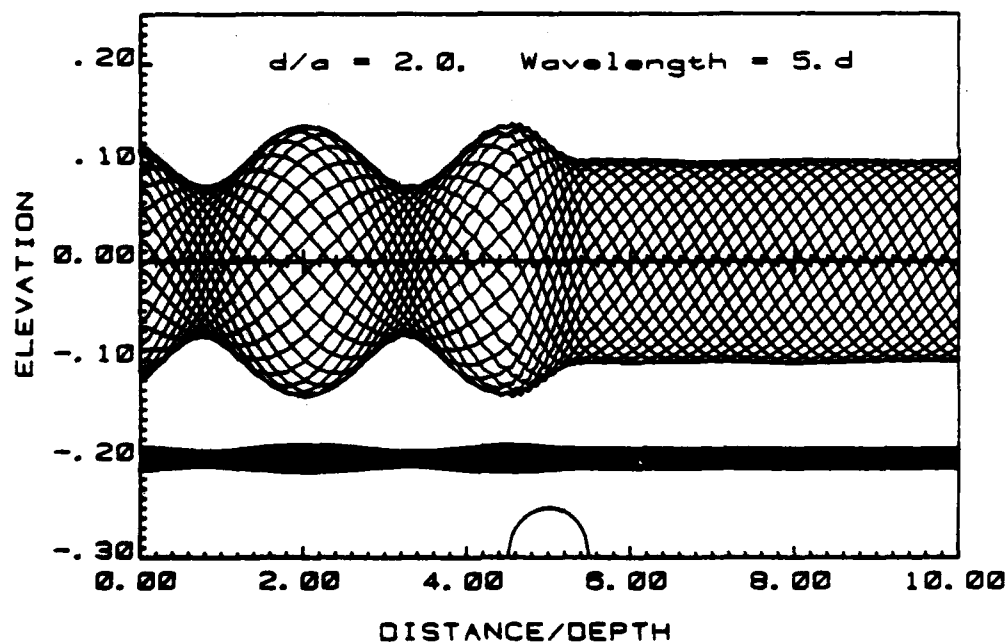


Fig. 4 — Surface elevation along channel during 25 cycles with half-cylinder on the bottom. Nonreflecting boundary at the right.

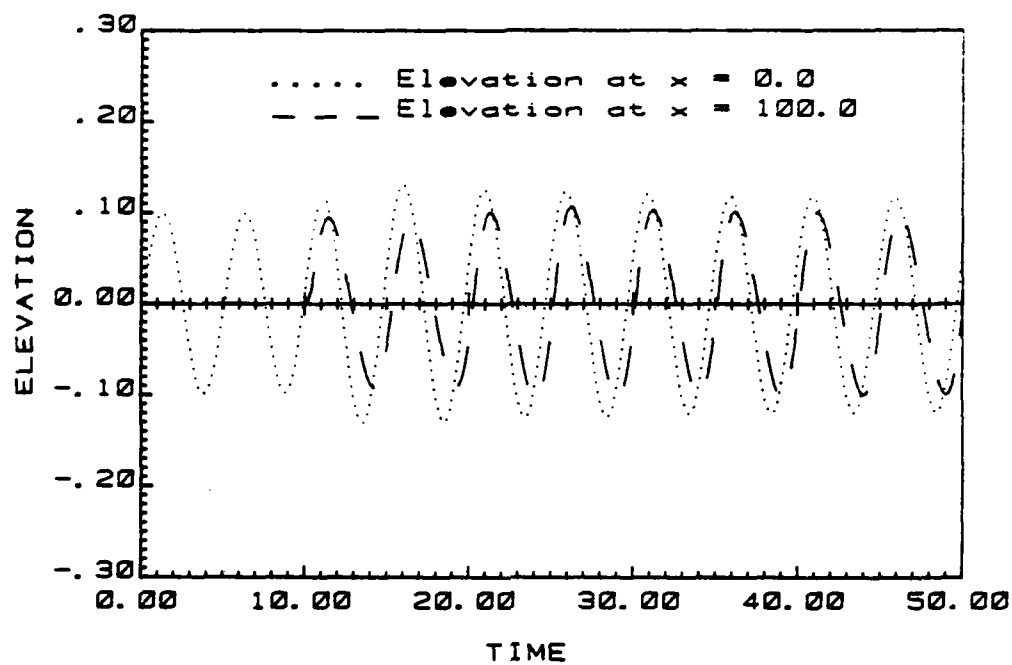


Fig. 5 — Time history of wave elevation at the left and right boundaries for the case shown in Fig. 4

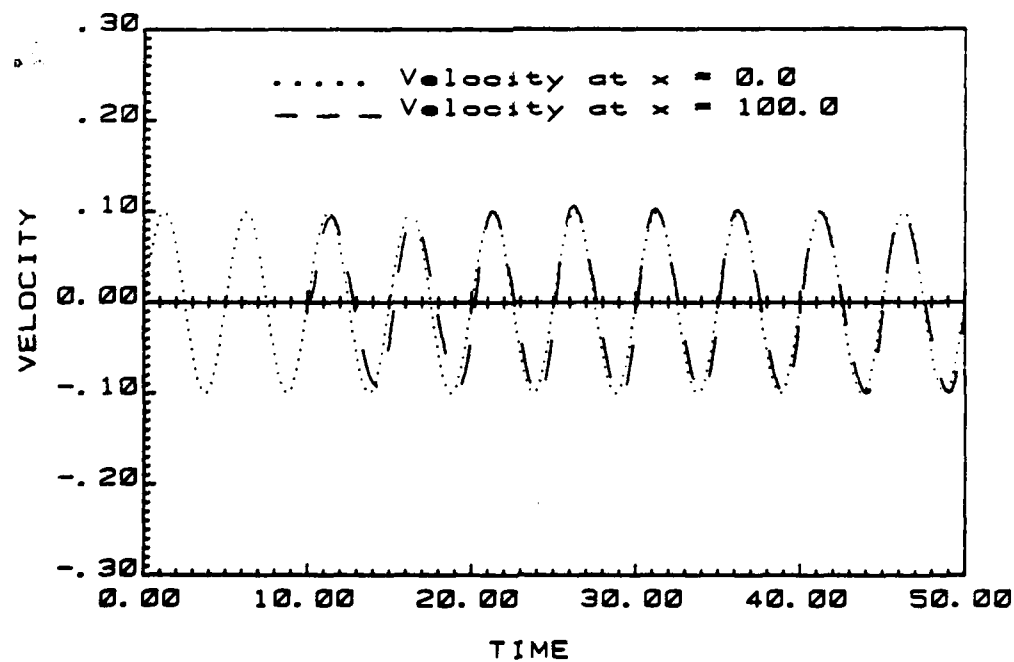


Fig. 6 — Time history of velocity at the left and right boundaries for the case shown in Fig. 4

```

0001
0002 *****
0003 *****
0004 *****
0005 ***** COMPUTATIONAL HYDRODYNAMICS *****
0006 *****
0007 ***** SIMULATION OF WATER WAVES *****
0008 ***** IN AN *****
0009 ***** OPEN CHANNEL *****
0010 *****
0011 *****
0012 *****
0013 ***** LU FOR INPUT OF DATA *****
0014 ***** LUIN # = 1, TERM IS THE INPUT DEVICE *****
0015 ***** LUIN # = 4, LEFT CASSETE IS THE INPUT DEV. *****
0016 ***** LUIN # = 5, RIGHT CASSETE IS THE INPUT DEV. *****
0017 ***** LUIN # = 8, TAPE DRIVE IS THE INPUT DEV. *****
0018 ***** TYPE THE LU # = 1 *****
0019 *****
0020 ***** INPUT NUMBER OF ELEMENTS. IF A HALF CYLINDER *****
0021 ***** IS GOING TO BE USED FOR THE CHANNEL GEOMETRY. *****
0022 ***** THE NUMBER OF ELEMENTS SHOULD BE A MULTIPLE OF *****
0023 ***** 20. NELEM = 100 *****
0024 *****
0025 ***** THE LENGTH OF EACH ELEMENT L IS DEFINED AS *****
0026 *****  $DX(L) = XL/NELEM$  *****
0027 ***** WHERE XL IS THE LENGTH OF THE CHANNEL *****
0028 *****
0029 *****
0030 ***** INPUT THE CHANNEL LENGTH, (XL = 100.0), XL = 100 *****
0031 *****
0032 ***** INPUT THE WATER DEPTH H0 = 10 *****
0033 *****
0034 ***** INPUT THE TIME STEP SIZE DT = .02 *****
0035 *****
0036 ***** INPUT THE MAX TIME FOR THE SIMULATION, TMAX = 1. *****
0037 *****
0038 ***** HOW FREQUENTLY WOULD YOU LIKE THE RESULTS *****
0039 ***** OUTPUTTED? IF IFREQ=10, THEN RESULTS WILL BE *****
0040 ***** OUTPUTTED EVERY 10 TIME STEPS. IFREQ = 10 *****
0041 *****
0042 ***** SPECIFY THE TYPE OF CHANNEL GEOMETRY *****
0043 ***** GEOM=0 CONSTANT BOTTOM SLOPE *****
0044 ***** GEOM=1 CHANNEL WITH A RAMP *****
0045 ***** GEOM=2 HALF-CYLINDER ON THE BOTTOM *****
0046 ***** GEOM=3 BOTTOM WITH CYLINDRICAL TRENCH *****
0047 ***** GEOM=4 BOTTOM SLOPED UPWARD *****
0048 ***** GEOM = 2 *****
0049 *****
0050 ***** SPECIFY THE TYPE OF CHANNEL *****
0051 ***** TYPE=1 INFINITE CHANNEL *****
0052 ***** TYPE=2 FINITE CHANNEL *****
0053 ***** TYPE = 1 *****
0054 *****
0055 ***** SPECIFY THE TYPE OF BOUNDARY CONDITION AT INLET*****
0056 ***** *****
0057 ***** IBC=1, H(1)=CONSTANT *****
0058 ***** IBC=2, H(1)=SINE WAVE *****

```

```

0059  ****      IBC=3,V(1)=CONSTANT      ****
0060  ****      IBC=4,V(1)=SINE WAVE      ****
0061  ****      IBC=5,H(1)=CONSTANT,FINITE CHANNEL      ****
0062  ****      IBC=6,H(1)=SINE WAVE,FINITE CHANNEL      ****
0063  ****      IBC = 2
0064  ****
0065  ****      IF YOU WANT TO INCLUDE THE BOTTOM FRICTION      ****
0066  ****      TERM INTO THE MOMENTUM EQUATION,      ****
0067  ****      TYPE 1 FOR YES, 0 FOR NO : 0
0068  ****
0069  ****      SPECIFY THE VALUE OF THE WAVE AMPLITUDE, AM0 = .1
0070  ****
0071  ****      SPECIFY THE VALUE OF THE WAVE PERIOD, T0 = 5
0072  ****
0073  ****      INPUT THE TIME TO START THE OUTPUT OF RESULT      ****
0074  ****      TSTAR = 0.
0075  ****
0076  ****      INPUT THE TIME TO STOP THE OUTPUT OF RESULTS      ****
0077  ****      TSTOP = 1.
0078  ****      TIME = 0.000
0079  ****      TIME = .200
0080  ****
0081  ****      DO YOU WANT TO OUTPUT THE RESULTS ON THE      ****
0082  ****      PRINTER? TYPE 1 FOR YES, 0 FOR NO : 0
0083  ****
0084  ****      DO YOU WANT TO OUTPUT THE RESULTS ON THE TAPE      ****
0085  ****      DRIVE? TYPE 1 FOR YES, 0 FOR NO : 0
0086  ****
0087  ****      SPECIFY WHICH OF THE FOLLOWING VARIABLES YOU      ****
0088  ****      WANT PLOTTED :
0089  ****      0 = NOTHING      ****
0090  ****      1 = ELEVATION      ****
0091  ****      2 = VELOCITY      ****
0092  ****      3 = PRESSURE      ****
0093  ****      4 = ELEVATION AND VELOCITY      ****
0094  ****      5 = VELOCITY AND PRESSURE      ****
0095  ****      6 = ELEVATION AND PRESSURE      ****
0096  ****      7 = ELEVATION,VELOCITY,AND PRESSURE      ****
0097  ****      IPLOT = 0
0098
0099  ****      TIME = .400
0100  ****      TIME = .600
0101  ****      TIME = .800
0102  ****      TIME = 1.000
0103
0104  ****      NUMBER OF POINTS, NP = 111
0105
0106
0107  ****      THE # OF POINTS FOR THE TIME PLOT IS = 5
0108
0109  ****
0110  ****      DO YOU WANT PRINTER OUTPUT OF THE NUMBER OF      ****
0111  ****      POINTS,PLOTS,ETC.? TYPE 1 FOR YES, 0 FOR NO : 0
0112  ****
0113  ****      DO YOU WANT TAPE OUTPUT OF ELEVATION AND      ****
0114  ****      VELOCITY? TYPE 1 FOR YES, 0 FOR NO : 0
0115  ****      READY TO PLOT GRAPHICS, PPESS RETURN
0116  ****
0117
0118  ****

```

```

0001 FTH7X,L
0002 PROGRAM KYMA(3,95), UPDATED 9:20:92
0003 C*****
0004 C*****
0005 C*****
0006 C*****THIS PROGRAM SOLVES THE DEPTH AVERAGED CONTINUITY *****
0007 C*****AND MOMENTUM EQUATIONS IN ONE DIMENSION. THE *****
0008 C*****COMPUTATIONAL MODEL IMPLEMENTED IN THIS PROGRAM *****
0009 C*****IS BASED ON THE FINITE ELEMENT METHOD. THE MODEL *****
0010 C*****IS DERIVED BY ASSUMING LINEAR APPROXIMATION IN *****
0011 C*****SPACE AND TIME. *****
0012 C***** FLUID DYNAMICS BRANCH *****
0013 C***** NAVAL RESEARCH LABORATORY *****
0014 C***** *****
0015 C*****
0016 C*****
0017 COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
0018 COMMON/BLOC2/ Q(222),YQ(222)
0019 COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PR(120)
0020 COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
0021 COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
0022 COMMON/BLOC6/ HTN(151),VTN(151),HT1(151),VT1(151),TIM(151)
0023 DIMENSION GSM(2505),IP(222)
0024 C*****
0025 WRITE(1,1000)
0026 1000 FORMAT("*****
0027 1 //, "***** *****"
0028 2 //, "***** *****"
0029 3 //, "***** COMPUTATIONAL HYDRODYNAMICS *****"
0030 4 //, "***** *****"
0031 5 //, "***** SIMULATION OF WATER WAVES *****"
0032 6 //, "***** IN AN *****"
0033 7 //, "***** OPEN CHANNEL *****"
0034 8 //, "***** *****"
0035 9 //, "*****" )
0036 GO TO 5
0037 100 CONTINUE
0038 WRITE(1,1040)
0039 WRITE(1,1005)
0040 1005 FORMAT("*****
0041 1 //, "**** DO YOU WANT TO RUN ANOTHER CASE? *****"
0042 2 //, "**** TYPE 1 FOR YES, 0 FOR NO : _")
0043 READ(1,*) ICASE
0044 IF(ICASE.LT.1) GO TO 9999
0045 WRITE(1,1010)
0046 1010 FORMAT("*****
0047 1 //, "**** WOULD YOU LIKE TO KEEP SOME OF THE DATA THE *****"
0048 2 //, "**** SAME? TYPE 1 FOR YES, 0 FOR NO : _")
0049 READ(LUIN,*) IQUES
0050 IF(IQUES.LT.1) ICALL=0
0051 IF(IQUES.LT.1) GO TO 5
0052 WRITE(1,1015)
0053 1015 FORMAT("*****
0054 1 //, "**** WHICH SET OF DATA WOULD YOU LIKE TO CHANGE? *****"
0055 2 //, "**** 0=MORE THAN ONE SET *****"
0056 3 //, "**** 1=LU # FOR INPUT OF DATA *****"
0057 4 //, "**** 2=INPUT DATA(CHANNEL LENGTH,TIME STEP,ETC.) *****"
0058 5 //, "**** 3=CHANNEL GEOMETRY *****"

```

```

0059      6      ,/, "**** WHICH SET? _" )
0060      READ(LUIN,*) ICALL
0061      IF(ICALL.GT.1) GO TO 6
0062      5 WRITE(1,1020)
0063      1020 FORMAT("*****")
0064      1      ,/, "***** LU FOR INPUT OF DATA *****"
0065      2      ,/, "**** LUIN # = 1, TERM IS THE INPUT DEVICE *****"
0066      3      ,/, "**** LUIN # = 4, LEFT CASSETE IS THE INPUT DEV. *****"
0067      4      ,/, "**** LUIN # = 5, RIGHT CASSETE IS THE INPUT DEV. *****"
0068      5      ,/, "**** LUIN # = 8, TAPE DRIVE IS THE INPUT DEV. *****"
0069      6      ,/, "**** TYPE THE LU # = _" )
0070      READ(1,*) LUIN
0071      IF(ICALL.EQ.1) GO TO 7
0072      C*****
0073      6      CALL INPUT(LUIN,ICALL)
0074      C*****
0075      7      IF(N.EQ.0) GO TO 9999
0076      C*****
0077      PI=3.14159
0078      GE=10.0      !GRAVITATIONAL ACCELERATION
0079      C0=SQRT(H0*GE) !WAVE VELOCITY
0080      C*****
0081      NUMEL=N      !NUMBER OF ELEMENTS
0082      NPOIN=N+1    !NUMBER OF NODAL POINTS
0083      NEQ=2*NPOIN !NUMBER OF EQUATIONS
0084      C*****
0085      C*****INITIALIZE PROGRAM PARAMETERS
0086      C*****
0087      T=0.0
0088      DO 10 I=1,NEQ
0089      IP(I)=0.0      !VECTOR FOR PIVOTING(DCOMP,SOLVE)
0090      Q(I)=0.0      !VECTOR OF NODAL UNKNOWN(S) (TIME T)
0091      YQ(I)=0.0     !VECTOR OF NODAL UNKNOWN(S) (TIME T-DT)
0092      10 CONTINUE
0093      DO 20 I=1,NPOIN
0094      V(I)=0.0      !VELOCITY VECTOR
0095      H(I)=0.0      !ELEVATION VECTOR
0096      RH(I)=HC(I)   !TOTAL WATER DEPTH VECTOR
0097      PR(I)=0.0     !PRESSURE VECTOR
0098      20 CONTINUE
0099      C*****
0100      NBAND=7      !BANDWIDTH OF MATRIX EQUATION
0101      NHBAN=(NBAND-1)/2
0102      JGF=NHBAN*NEQ
0103      JGSM=JGF+NEQ
0104      JEND=JGSM+NEQ*NBAND
0105      IF(JEND.GT.2505) GO TO 9999
0106      ICUNTE=0
0107      IT=0
0108      IPT=0
0109      NPL0T=0
0110      C*****
0111      WRITE(1,1025)
0112      1025 FORMAT("*****")
0113      1      ,/, "**** INPUT THE TIME TO START THE OUTPUT OF RESULT *****"
0114      2      ,/, "**** TSTAR = _" )
0115      READ(LUIN,*) TSTAR
0116      WRITE(1,1030)
0117      1030 FORMAT("*****")
0118      1      ,/, "**** INPUT THE TIME TO STOP THE OUTPUT OF RESULTS *****"

```

```

0119      2      ,/, "**** TSTOP = _" )
0120      READ(LUIN,*) TSTOP
0121      150 CONTINUE
0122      WRITE(1,1035) T
0123      IF(T.LE.TSTAR.OR.T.GE.TSTOP) GO TO 200
0124 C*****
0125      CALL XOUT(LUIN,IPT)
0126 C*****
0127 C*****STORE H(0,T),H(L,T),V(0,T),V(L,T)
0128 C*****
0129      IT=IT+1
0130      HTN(IT)=H(NPOIN)      IM(L,T)
0131      VTN(IT)=V(NPOIN)      IV(L,T)
0132      HT1(IT)=H(1)      IM(0,T)
0133      VT1(IT)=V(1)      IV(0,T)
0134      TIM(IT)=T
0135      IF(IT.GE.151) GO TO 46
0136 C*****
0137      200      T=T+DT
0138 C*****
0139 C*****THE ARRAY GSM(I) IS USED FOR STORING THE NBAND
0140 C*****DIAGONALS OF THE STIFFNESS MATRIX
0141 C*****GSM(I), I=JGF+1,JGSM, RIGHT HAND SIDE VECTOR OR
0142 C*****      SOLUTION VECTOR.
0143 C*****GSM(I), I=JGSM+1,JEND, 1ST,2ND,...,7TH NON-
0144 C*****      ZERO GIAGONALS.
0145 C*****
0146      DO 30 I=1,JEND
0147      30      GSM(I)=0.0
0148 C*****
0149      CALL QSOLV(GSM(1),JEND,IP,NEG,NBAND)
0150 C*****
0151      DO 40 I=1,NEG
0152      Q(I)=GSM(I+JGF)
0153      40 CONTINUE
0154 C*****
0155 C*****COMPUTE THE ELEVATION, VELOCITY & PRESSURE
0156 C*****
0157      DO 45 I=1,NPOIN
0158      H(I)=Q(2*I-1)
0159      V(I)=Q(2*I)
0160      RH(I)=HC(I)+H(I)
0161      PR(I)=H(I)+0.5*V(I)**2
0162      45 CONTINUE
0163 C*****
0164 C*****IF T EXCEEDS THAX, TERMINATE INTEGRATION.....
0165 C*****
0166      IF(T.LE.THAX) GO TO 50
0167 C*****
0168      GO TO 46
0169      46 WRITE(1,47)
0170      47 FORMAT("***** WARNING *****")
0171      1      ,/, "**** THE NUMBER OF POINTS FOR THE TIME PLOTS HAS ****"
0172      2      ,/, "**** EXCEEDED THE MAXIMUM NUMBER OF POINTS ALLOWED. ****"
0173      3      ,/, "**** INCREASE IFREQ FOR LESS TIME PLOTS IF ANOTHER ****"
0174      4      ,/, "**** CASE IS DESIRED. ****"
0175      5      ,/, "*****" )
0176 C*****
0177      48 CALL TOUT(LUIN,NPOIN,IT,IPT)
0178 C*****

```



```

0238 1025 FORMAT("*****")
0239 1 //,"**** INPUT THE WATER DEPTH H0 = _"
0240 READ(LUIN,*) H0
0241 WRITE(1,1030)
0242 1030 FORMAT("*****")
0243 1 //,"**** INPUT THE TIME STEP SIZE DT = _"
0244 READ(LUIN,*) DT
0245 WRITE(1,1035)
0246 1035 FORMAT("*****")
0247 1 //,"**** INPUT THE MAX TIME FOR THE SIMULATION, TMAX = _"
0248 READ(LUIN,*) TMAX
0249 WRITE(1,1040)
0250 1040 FORMAT("*****")
0251 1 //,"**** HOW FREQUENTLY WOULD YOU LIKE THE RESULTS *****"
0252 2 //,"**** OUTPUTTED? IF IFREQ=10, THEN RESULTS WILL BE *****"
0253 3 //,"**** OUTPUTTED EVERY 10 TIME STEPS. IFREQ = _"
0254 READ(LUIN,*) IFREQ
0255 C*****
0256 C*****SET CHANNEL GEOMETRY

0257 C*****
0258 30 CALL GEOMT(NPOIN,LUIN,IT,ICALL)
0259 C*****
0260 IF(ICALL.EQ.3) GO TO 40
0261 WRITE(1,1045)
0262 1045 FORMAT("*****")
0263 1 //,"**** SPECIFY THE TYPE OF BOUNDARY CONDITION AT INLET*****"
0264 2 //,"**** *****"
0265 3 //,"**** IBC=1,H(1)=CONSTANT *****"
0266 4 //,"**** IBC=2,H(1)=SINE WAVE *****"
0267 5 //,"**** IBC=3,V(1)=CONSTANT *****"
0268 6 //,"**** IBC=4,V(1)=SINE WAVE *****"
0269 7 //,"**** IBC=5,H(1)=CONSTANT,FINITE CHANNEL *****"
0270 8 //,"**** IBC=6,H(1)=SINE WAVE,FINITE CHANNEL *****"
0271 9 //,"**** IBC = _"
0272 READ(LUIN,*) IBC
0273 40 CONTINUE
0274 NUMBC=2 ! NUMBER OF BOUNDARY CONDITIONS REQUIRED
0275 GO TO (50,50,51,51,50,50), IBC
0276 50 IB(1)=1 ! INDEX FOR BOUNDARY CONDITION AT X=0
0277 IB(2)=2*(N+1) ! INDEX FOR BOUNDARY CONDITION AT X=100
0278 GO TO 57
0279 51 IB(1)=2
0280 IB(2)=2*(N+1)-1
0281 57 IF(ICALL.EQ.3) GO TO 60
0282 WRITE(1,1050)
0283 1050 FORMAT("*****")
0284 1 //,"**** IF YOU WANT TO INCLUDE THE BOTTOM FRICTION *****"
0285 2 //,"**** TERM INTO THE MOMENTUM EQUATION, *****"
0286 3 //,"**** TYPE 1 FOR YES, 0 FOR NO : _"
0287 READ(LUIN,*) ICZ
0288 IF(ICZ.LE.0) GO TO 59
0289 WRITE(1,1055)
0290 1055 FORMAT("*****")
0291 1 //,"****SPECIFY THE VALUE OF THE CHEZY COEFFICIENT. CZ = _"
0292 READ(LUIN,*) CZ
0293 59 WRITE(1,1060)
0294 1060 FORMAT("*****")
0295 1 //,"****SPECIFY THE VALUE OF THE WAVE AMPLITUDE. AMO = _"
0296 READ(LUIN,*) AMO
0297 GO TO (60,70,60,70,60,70), IBC
0298 70 WRITE(1,1065)

```

```

0299 1065 FORMAT("*****")
0300 1 ,/, "****SPECIFY THE VALUE OF THE WAVE PERIOD. T0 = _")
0301 READ(LUIN,*) T0
0302 90 RETURN
0303 END
0304 C*****

```

&KYMA T=00003 IS ON CR00011 USING 00168 BLKS R=0000

```

0305 C*****
0306 C*****
0307 BLOCK DATA
0308 COMMON/BLOC1/ T0,T,TMAX,IFREQ,ISC,NUMBC,IB(2)
0309 COMMON/BLOC2/ Q(222),YQ(222)
0310 COMMON/BLOC3/ HK(120),VK(120),DH(120),DV(120),PP(120)
0311 COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
0312 COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
0313 COMMON/BLOC6/ HTN(151),VTN(151),HTI(151),VTI(151),TIM(151)
0314 END
0315 C*****

```

&KYMA T=00003 IS ON CR00011 USING 00168 BLKS R=0000

```

0316 C*****
0317 C*****
0318 SUBROUTINE GEONT(NPOIN,LUIN,IT,ICALL), N.C.CHU, G.A.KERAMIDAS
0319 C*****
0320 COMMON/BLOC3/ HK(120),VK(120),DH(120),DV(120),PR(120)
0321 COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
0322 COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
0323 C*****
0324 IF(ICALL.EQ.2) GO TO 5
0325 WRITE(1,1000)
0326 1000 FORMAT("*****")
0327 1 ,/, "**** SPECIFY THE TYPE OF CHANNEL GEOMETRY *****"
0328 2 ,/, "**** GEOM=0 CONSTANT BOTTOM SLOPE *****"
0329 3 ,/, "**** GEOM=1 CHANNEL WITH A RAMP *****"
0330 4 ,/, "**** GEOM=2 HALF-CYLINDER ON THE BOTTOM *****"
0331 5 ,/, "**** GEOM=3 BOTTOM WITH CYLINDRICAL TRENCH *****"
0332 6 ,/, "**** GEOM=4 BOTTOM SLOPED UPWARD *****"
0333 7 ,/, "**** GEOM = _" )
0334 READ(LUIN,*) IGEOM
0335 WRITE(1,1010)
0336 1010 FORMAT("*****")
0337 1 ,/, "**** SPECIFY THE TYPE OF CHANNEL *****"
0338 2 ,/, "**** TYPE=1 INFINITE CHANNEL *****"
0339 3 ,/, "**** TYPE=2 FINITE CHANNEL *****"
0340 4 ,/, "**** TYPE = _" )
0341 READ(LUIN,*) ITYPE
0342 IF(ITYPE.EQ.1) GO TO 5
0343 WRITE(1,1015)
0344 1015 FORMAT("*****")
0345 1 ,/, "**** INPUT THE DEPTH OF THE CHANNEL AT THE CLOSED *****"
0346 2 ,/, "**** END. HNO = _" )
0347 READ(LUIN,*) HNO
0348 GO TO 6
0349 5 CONTINUE
0350 IF(IGEOM.NE.4) GO TO 6
0351 WRITE(1,1020)

```

```

0352 1020 FORMAT("*****")
0353 1 //,"*** INPUT THE DEPTH OF THE WATER AT THE END OF *****"
0354 2 //,"*** THE CHANNEL DWO = _"
0355 READ(LUIN,*) DWO
0356 HNO=H0-DWO
0357 6 DX0=1.0/W0 ! CONSTANT ELEMENT LENGTH = XL/N
0358 DO 7 I=1,NPOIN
0359 DX(I)=DX0 ! ELEMENT LENGTH VECTOR
0360 HC(I)=H0 ! WATER DEPTH IN CHANNEL
0361 RL(I)=DX(I)*FLOAT(I-1) ! NODAL POINTS COORDINATES
0362 7 CONTINUE
0363 IF(ITYPE.EQ.1) GO TO 9
0364 HC(NPOIN)=HNO
0365 9 IF(IGEOM.LE.0) GO TO 50
0366 GO TO (11,21,31,39), IGEOM
0367 C*****
0368 C*****GEOMETRY FOR CHANNEL WITH A RAMP
0369 C*****
0370 11 CONTINUE
0371 HNO=5
0372 NEH=N/2+1
0373 NEC=N/10
0374 NED=NEH+NEC
0375 DO 15 I=NEH,NED
0376 DX(I)=DX0/2.0
0377 15 HC(I)=H0-FLOAT(I-NEH)*DX(I)
0378 NMAX=NPOIN+NEC
0379 DO 20 I=NED,NMAX
0380 DX(I)=DX0
0381 20 HC(I)=H0-HNO
0382 GO TO 43
0383 C*****
0384 C*****GEOMETRY FOR CHANNEL WITH A HALF-CYLINDER ON THE BOTTOM
0385 C*****
0386 21 NEC=N/10
0387 NC0=(N-NEC)/2
0388 NC1=NC0+1
0389 NC2=NC0+2*NEC
0390 DR=XL/10.0
0391 RR=DR/2.0
0392 C*****
0393 DO 25 I=1,NC0
0394 DX(I)=DX0
0395 DX(I+NC2)=DX0
0396 25 CONTINUE
0397 C*****
0398 DO 27 I=1,NPOIN+NEC
0399 HC(I)=H0
0400 27 CONTINUE
0401 C*****
0402 DO 30 I=NC1,NC2
0403 I1=I-(NEC/2+1)
0404 DX(I)=DX0/2.0
0405 XR=-XL/4.0+DX(I)*FLOAT(I1)
0406 HC(I)=H0-SQRT(RR**2-XR**2)
0407 30 CONTINUE
0408 C*****
0409 GO TO 43
0410 C*****
0411 C*****GEOMETRY FOR A CHANNEL WITH A CYLINDRICAL TRENCH ON THE BOTTOM
0412 C*****

```

```

0413      31      NEC=N/10
0414      NC0=(N-NEC)/2
0415      NC1=NC0+1
0416      NC2=NC0+2+NEC
0417      DR=XL/10.0
0418      RR=DR/2.0
0419      C*****
0420      DO 36 I=1,NC0
0421          DX(I)=DX0
0422          DX(I+NC2)=DX0
0423      36 CONTINUE
0424      C*****
0425      DO 37 I=1,NPOIN+NEC
0426          HC(I)=H0
0427      37 CONTINUE
0428      C*****
0429      DO 38 I=NC1,NC2
0430          II=I-(NEC/2+1)
0431          DX(I)=DX0/2.0
0432          XR=-XL/4.0+DX(I)*FLOAT(II)
0433          HC(I)=H0+SQRT(RR**2-XR**2)
0434      38 CONTINUE
0435      GO TO 43
0436      C*****
0437      C*****GEOMETRY FOR A CHANNEL WITH THE BOTTOM SLOPED UPWARD
0438      C*****
0439      39      SLOPE=HN0/XL
0440      DO 41 I=1,NPOIN
0441          HC(I)=H0-DX(I)*SLOPE*FLOAT(I-1)
0442      41 CONTINUE
0443      C*****
0444      43      N=N+NEC      ! THE NUMBER OF ELEMENTS HAS BEEN INCREASED
0445      C      ! BY NEC FOR THE SPECIFIED GEOMETRY
0446      DO 45 I=1,N
0447          RL(I+1)=RL(I)+DX(I)
0448      45 CONTINUE
0449      C*****
0450      50 CONTINUE
0451      RETURN
0452      END
0453      C*****

```

SKYNA T=00003 IS ON CR00011 USING 00160 BLKS R=0000

```

0454      C*****
0455      C*****
0456      SUBROUTINE ELMAT(L,ESM1,ESM2), G.A.KERANIDAS
0457      C*****
0458      C*****SUBROUTINE ELMAT EVALUATES THE ELEMENT MATRICES
0459      C*****
0460      COMMON/BLOC1/ T0,T,THAX,IFREQ,IBC,NUMBC,IB(2)
0461      COMMON/BLOC3/ HK(120),VK(120),DH(120),DV(120),PR(120)
0462      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
0463      COMMON/BLOC5/ N,H0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
0464      DIMENSION AK(4,4),B(4,4),ESM1(4,4),ESM2(4,4)
0465      C*****
0466      C*****MASS MATRIX, CORRESPONDS TO MATRIX [M] OF EQ. (3)
0467      C*****
0468      AK(1,1)=2./DT
0469      AK(1,2)=0.0
0470      AK(1,3)=1./DT
0471      AK(1,4)=0.0

```

```

0472      A(2,1)=0.0
0473      A(2,2)=2./DT
0474      A(2,3)=0.0
0475      A(2,4)=1./DT
0476      A(3,1)=1./DT
0477      A(3,2)=0.0
0478      A(3,3)=2./DT
0479      A(3,4)=0.0
0480      A(4,1)=0.0
0481      A(4,2)=1./DT
0482      A(4,3)=0.0
0483      A(4,4)=2./DT
0484      C*****
0485      C*****CONVECTIVE MATRIX, CORRESPONDS TO MATRIX [K] OF EQ. (3)
0486      C*****
0487          C0=SQRT(GE*RH(L))
0488          DV0=V(L)-V(L+1)
0489          DIF0=SQRT(2.*DX(L)/(C0*DT))
0490          DIF1=1./DIF0
0491          DIF2=DIF0**2
0492          DIFL=(DIF2*ABS(DV0)+C0*DIF1)/DX(L)
0493      C*****
0494          HL0=0.5*(HC(L)+HC(L+1))
0495          HL1=(2.*HC(L)+HC(L+1))/DX(L)
0496          HL2=(HC(L)+2.*HC(L+1))/DX(L)
0497          DHDX=(HC(L+1)-HC(L))/DX(L)
0498          DVDX=(V(L+1)-V(L))/DX(L)
0499          DHC=(HC(L+1)-HC(L))/DX(L)
0500          AVEL=0.5*(V(L)+V(L+1))
0501          ADEP=0.5*(RH(L)+RH(L+1))
0502          DISP=0.0
0503          IF(ICZ.GT.0) DISP=GE/(ADEP+CZ**2)
0504          DIS11=DISP*(V(L)+AVEL)
0505          DIS22=DISP*(V(L+1)+AVEL)
0506          DIS12=DISP*AVEL
0507          DIS21=DISP*AVEL
0508      C*****
0509          B(2,2)=2.*DVDX+DIFL+DIS11
0510          B(2,1)=3.*GE/DX(L)
0511          B(2,4)=DVDX-DIFL+DIS12
0512          B(2,3)=3.*GE/DX(L)
0513          B(1,2)=2.*(DHDX+DHC)-HL1
0514          B(1,1)=2.*DVDX
0515          B(1,4)=DHDX+DHC+HL1
0516          B(1,3)=DVDX
0517          B(4,2)=DVDX-DIFL+DIS21
0518          B(4,1)=-3.*GE/DX(L)
0519          B(4,4)=2.*DVDX+DIFL+DIS22
0520          B(4,3)=-3.*GE/DX(L)
0521          B(3,2)=DHDX+DHC-HL2
0522          B(3,1)=DVDX
0523          B(3,4)=2.*(DHDX+DHC)+HL2
0524          B(3,3)=2.*DVDX
0525      C*****
0526          DO 10 I=1,4
0527          DO 10 J=1,4
0528      C*****
0529      C*****ELEMENT STIFFNESS MATRICES,
0530      C*****
0531          ESM1(I,J)=A(I,J)+0.5*B(I,J)      !MATRIX [A] OF EQ.(4)
0532          ESM2(I,J)=A(I,J)-0.5*B(I,J)      !MATRIX [B] OF EQ.(4)
0533      10 CONTINUE
0534      C*****

```

```

0535      RETURN
0536      END
0537 C*****

```

&KYMA T=00003 IS ON CR00011 USING 00168 BLKS R=0000

```

0538 C*****
0539 C*****
0540      SUBROUTINE QSOLV(GSM,JEND,IP,NEQ,NBAND), G.A.KERAMIDAS
0541 C*****
0542 C*****SUBROUTINE QSOLV FORMS THE SYSTEM OF ALGEBRAIC EQUATIONS
0543 C*****AND CALLS SUBROUTINE MPART TO PARTITION THE EQUATIONS ACCORDING
0544 C*****TO THE BOUNDARY CONDITIONS. IT ALSO CALLS SUBROUTINES DCOMP
0545 C*****AND SOLVE FOR THE SOLUTION OF THE SYSTEM OF EQUATIONS.
0546 C*****
0547      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
0548      COMMON/BLOC2/ Q(222),YQ(222)
0549      COMMON/BLOC3/ HK(120),V(120),DH(120),DV(120),PR(120)
0550      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
0551      COMMON/BLOC5/ N,M0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
0552      DIMENSION GSM(JEND),IP(NEQ)
0553      DIMENSION ESM1(4,4),ESM2(4,4),EF(4),NS(4),BV(2)
0554 C*****
0555      NUMEL=N
0556      NPOIN=N+1
0557      NNBAN=(NBAND-1)/2      ! HALF BANDWIDTH
0558      NORD=2                  ! DEGREES OF FREEDOM PER NODE
0559      NDOFR=2*NORD           ! DEGREES OF FREEDOM PER ELEMENT
0560      JGF=NNBAN+NEQ
0561      JGSM=JGF+NEQ
0562      PI=3.14159
0563      C0=SQRT(H0*GE)
0564 C*****
0565      DO 10 I=1,4
0566      DO 10 J=1,4
0567          ESM1(I,J)=0.0
0568      10  ESM2(I,J)=0.0
0569      DO 15 I=1,NDOFR
0570      15  NS(I)=I-NORD
0571 C*****
0572      DO 20 I=1,JEND
0573      20  GSM(I)=0.0
0574 C*****
0575 C*****BOUNDARY CONDITIONS
0576 C*****
0577      GO TO (21,22,23,24,25,26), IBC
0578      21      HK(1)=AM0
0579          Q(1)=HK(1)
0580          V(NPOIN)=H(NPOIN)*SQRT(GE*HC(NPOIN))/HC(NPOIN)
0581          Q(NEQ)=V(NPOIN)
0582      GO TO 30
0583      22      HK(1)=AM0*SIN(2.*PI*T/T0)
0584          Q(1)=HK(1)
0585          V(NPOIN)=H(NPOIN)*SQRT(GE*HC(NPOIN))/HC(NPOIN)
0586          Q(NEQ)=V(NPOIN)
0587      GO TO 30
0588      23      V(1)=AM0
0589          Q(2)=V(1)
0590          H(NPOIN)=V(NPOIN)*HC(NPOIN)/SQRT(GE*HC(NPOIN))
0591          Q(NEQ-1)=H(NPOIN)
0592      GO TO 30
0593      24      V(1)=AM0*SIN(2.*PI*T/T0)
0594          Q(2)=V(1)
0595          H(NPOIN)=V(NPOIN)*HC(NPOIN)/SQRT(GE*HC(NPOIN))

```

```

0596      Q(NEQ-1)=H(NPOIN)
0597      GO TO 30
0598      25      H(1)=AM0
0599              Q(1)=H(1)
0600              V(NPOIN)=0.0
0601      Q(NEQ)=0.0
0602      GO TO 30
0603      26      H(1)=AM0*SIN(2.*PI*T/T0)
0604              Q(1)=H(1)
0605              V(NPOIN)=0.0
0606      Q(NEQ)=0.0
0607      30 CONTINUE
0608      BV(1)=Q(IB(1))
0609      BV(2)=Q(IB(2))
0610      C*****
0611      C*****GENERATION OF THE BOUNDARY FORCE VECTOR
0612      C*****
0613      DO 45 LL=1,NUMEL
0614          EF(1)=0.00
0615          EF(2)= 6.*GE*H(LL)/DX(LL)
0616          EF(3)=0.0
0617          EF(4)=-6.*GE*H(LL+1)/DX(LL)
0618      C*****
0619      NS(1)=NS(1)+NORD
0620      NS(2)=NS(2)+NORD
0621      NS(3)=NS(3)+NORD
0622      NS(4)=NS(4)+NORD
0623      DO 40 I=1,NDOFR
0624          II=NS(I)
0625          I2=JGF+II
0626      40      GSM(I2)=GSM(I2)+EF(I)      ! BOUNDARY FORCE VECTOR
0627      45      CONTINUE
0628      C*****
0629      DO 35 I=1,NDOFR
0630      35      NS(I)=I-NORD
0631      C*****
0632      C*****GENERATION OF THE SYSTEM MATRICES
0633      C*****
0634      DO 80 KK=1,NUMEL
0635      CALL ELMAT(KK,ESM1,ESM2)
0636      NS(1)=NS(1)+NORD
0637      NS(2)=NS(2)+NORD
0638      NS(3)=NS(3)+NORD
0639      NS(4)=NS(4)+NORD
0640      DO 80 I=1,NDOFR
0641          II=NS(I)
0642          J2=JGF+II
0643      DO 75 J=1,NDOFR
0644          JJ=NS(J)
0645      GSM(J2)=GSM(J2)+ESM2(I,J)*YQ(JJ)      ! RIGHT HAND SIDE VECTOR
0646          JJ=JJ-II+NMBAN
0647          JS=JGSM+JJ+NEQ+II
0648      GSM(JS)=GSM(JS)+ESM1(I,J)      ! MATRIX COEFFICIENT OF SYSTEM
0649      75      CONTINUE
0650      80      CONTINUE
0651      C*****
0652      CALL MPART(GSM(JGSM+1),GSM(JGF+1),NEQ,BV,IB,NBAND,NUMBC)
0653      CALL DCOMP(GSM(1),IP,GSM(JGSM+1),NEQ,NEQ,NMBAN,NBAND)
0654      CALL SOLVE(GSM(1),IP,GSM(JGF+1),GSM(JGSM+1),NEQ,NEQ,NMBAN,NBAND)
0655      C*****
0656      RETURN
0657      END
0658      C*****

```

SKYMA T=00003 IS ON CR00011 USING 00168 BLKS R=0000

```
0659 C+*****
0660 C+*****
0661 SUBROUTINE XPLOT(R,T,N,IPT,IPLT,ICPL), N.C.CHU, G.A.KERAMIDAS
0662 C+*****
0663 C+*****SUBROUTINE XPLOT PLOTS THE SOLUTION ACCORDING TO THE X-COORDINATE
0664 C+*****
0665 COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
0666 COMMON/BLOC5/ NP,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
0667 DIMENSION R(N),H(200),X(120),Y(200)
0668 C+*****
0669 DO 10 I=1,N
0670 X(I)=RL(I)/10.0
0671 Y(I)=(H0-HC(I))/100.0-0.30
0672 H(I)=R(I)/10.0-(0.3-H0/100.)
0673 10 CONTINUE
0674 C+*****
0675 IF(IPLT.GT.3) GO TO 979
0676 IF(ICPL.GT.0) GO TO 47
0677 C+*****PLOT GRAPHICS OUTPUT
0678 979 WRITE(1,980)
0679 980 FORMAT("READY TO PLOT GRAPHICS, PRESS RETURN")
0680 READ(1,*) IIC
0681 IF(IGEOM.NE.3) GO TO 15
0682 989 FORMAT("E*dF E*dA E*a0.110.m1.0p0.2q-.3n.25o.05r.01s1vC")
0683 WRITE(1,981)
0684 981 FORMAT("E*dF E*dA E*a0.110.m1.0p0.2q-.35n.25o.05r.01s1vC")
0685 GO TO 20
0686 15 WRITE(1,989)
0687 20 WRITE(1,983)
0688 983 FORMAT("E*a2h1i2j1K")
0689 WRITE(1,982)
0690 982 FORMAT("E*d300.0okS E*a2m1N E+1X-AXIS E+dT")
0691 IF(IPT.EQ.0) GO TO (30,40,45,30,40,30,30), IPLT
0692 IF(IPT.EQ.1) GO TO (47,47,47,40,45,45,40), IPLT
0693 IF(IPT.EQ.2) GO TO (47,47,47,47,47,47,45), IPLT
0694 30 WRITE(1,903)
0695 GO TO 47
0696 40 WRITE(1,913)
0697 GO TO 47
0698 45 WRITE(1,923)
0699 903 FORMAT("E*d30.160okS E*a2m2N E+1ELEVATION E+dT")
0700 913 FORMAT("E*d30.160okS E*a2m2N E+1VELOCITY E+dT")
0701 923 FORMAT("E*d30.160okS E*a2m2N E+1PRESSURE E+dT")
0702 47 WRITE(1,984)
0703 984 FORMAT("E*aA")
0704 WRITE(1,987)
0705 987 FORMAT("E*dF")
0706 C+*****PLOT COMPUTED SOLUTION AT TIME T
0707 WRITE(1,985) (X(I),R(I),I=1,N)
0708 WRITE(1,985) (X(N-I+1),H(N-I+1),I=1,N)
0709 IF(IPT.GT.1) GO TO 48
0710 WRITE(1,985) (X(I),Y(I),I=1,N)
0711 985 FORMAT(2F7.4)
0712 48 WRITE(1,988)
0713 988 FORMAT("E*dE")
0714 C+*****
0715 WRITE(1,986)
0716 986 FORMAT("E*aB E+dT E+dE")
```

```

0717      WRITE(1,997)
0718      WRITE(1,998) T
0719      997 FORMAT('E+d140,310okS E+m2m1N E+1E M E+dT E+m1m1N')
0720      998 FORMAT('E+d140,310okS E+m2m1N E+1 T =',F5.3,'E+dT E+m1m1N')
0721      50 RETURN
0722      END
0723 C*****

```

&KYMA T=00003 IS ON CR00011 USING 00168 BLKS R=0000

```

0724 C*****
0725 C*****
0726      SUBROUTINE TPLOT(R,TIM,N,IPT), G.A.KERAMIDAS
0727 C*****
0728 C*****SUBROUTINE TPLOT PLOTS THE SOLUTION ACCORDING TO THE TIME COORDINATE
0729 C*****
0730      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
0731      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
0732      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
0733      DIMENSION R(N),TIM(N)
0734 C*****
0735 C*****
0736 C. . . PLOT GRAPHICS OUTPUT . . . . .
0737      WRITE(1,980)
0738      980 FORMAT('READY TO PLOT GRAPHICS, PRESS RETURN')
0739      READ(1,*) II
0740      WRITE(1,981)
0741      981 FORMAT('E+dF E+dA E+a50.0m5,0p1qC')
0742      WRITE(1,982)
0743      982 FORMAT('E+d300,0okS E+m2m1N E+1T-AXIS E+dT')
0744      IF(IPT.EQ.0) WRITE(1,903)
0745      IF(IPT.EQ.1) WRITE(1,913)
0746      IF(IPT.EQ.2) WRITE(1,923)
0747      903 FORMAT('E+d30,160okS E+m2m2N E+1ELEVATION E+dT')
0748      913 FORMAT('E+d30,160okS E+m2m2N E+1VELOCITY E+dT')
0749      923 FORMAT('E+d30,160okS E+m2m2N E+1ELEVATION E+dT')
0750      WRITE(1,984)
0751      984 FORMAT('E+aA')
0752 C. . . PLOT COMPUTED SOLUTION AT TIME T1 . . . . .
0753      WRITE(1,985) (TIM(I),R(I),I=1,N)
0754      985 FORMAT(2F7.4)
0755      WRITE(1,986)
0756      986 FORMAT('E+aB E+dT E+dE')
0757      WRITE(1,998) T
0758      998 FORMAT('E+d140,310okS E+m2m1N E+1 T =',F4.2,'E+dT E+m1m1N')
0759      RETURN
0760      END
0761 C*****

```

&KYMA T=00003 IS ON CR00011 USING 00168 BLKS R=0000

```

0762 C*****
0763 C*****
0764      SUBROUTINE XOUT(LUIN,IPT), N.C.CHU
0765 C*****
0766      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
0767      COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PP(120)
0768      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
0769      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
0770      COMMON/BLOC6/ HTN(151),VTN(151),HT(151),VT(151),TIM(151)
0771 C*****

```

```

0772      NPOIN=N+1
0773      IF(IPT.GT.0) GO TO 20
0774      ICPL=0
0775      ICPL0=0
0776      WRITE(1,1000)
0777 1000 FORMAT("*****")
0778      1 //,"**** DO YOU WANT TO OUTPUT THE RESULTS ON THE *****"
0779      2 //,"**** PRINTER? TYPE 1 FOR YES, 0 FOR NO : _")
0780      READ(LUIN,*) IPRIN
0781      IF(IPRIN.LT.1) GO TO 30
0782      WRITE(6,1035)
0783      WRITE(6,1005)
0784 1005 FORMAT(40X,"*****")
0785      1**",//,40X,"****"
0786      2* ",//,40X,"****"
0787      3* ",//,40X,"****" COMPUTATIONAL HYDRODYNAMICS
0788      4* ",//,40X,"****"
0789      5* ",//,40X,"****" SIMULATION OF WATER WAVES
0790      6* ",//,40X,"****" IN AN
0791      7* ",//,40X,"****" OPEN CHANNEL
0792      8* ",//,40X,"****"
0793      9* ",//,40X,"*****"
0794      1* ")
0795      C0=SQRT(H0*GE)
0796      CRO=C0*DT*W0
0797      WRITE(6,1060)
0798      WRITE(6,1010) CRO
0799 1010 FORMAT(40X,"*****")
0800      1**",//,40X,"****"
0801      2* ",//,40X,"****" THE COURANT NUMBER IS ",F6.3,11X,"****"
0802      3 //, 40X,"****"
0803      4* ",//,40X,"*****"
0804      5* ")
0805      WRITE(6,1060)
0806      WRITE(6,1015) N,XL,DX(1),DT,AM0,IBC
0807 1015 FORMAT(40X,"*****")
0808      1**",//,40X,"****"
0809      2* ",//,40X,"****" NUMBER OF ELEMENTS IS ",I6,10X,"****"
0810      3 //,40X,"****" LENGTH OF CHANNEL IS ",F6.3,11X,"****"
0811      4 //,40X,"****" LENGTH OF ELEMENT IS ",F6.3,11X,"****"
0812      5 //,40X,"****" TIME STEP SIZE IS ",F6.3,14X,"****"
0813      6 //,40X,"****" AMPLITUDE OF INCOMING WAVE IS ",F6.3,2X,"
0814      7****",//,40X,"****" BOUDARY CONDITION AT X=0.0 IS IBC=",I2,
0815      92X,"****",//,40X,"****" IBC=1,H(1)=CONSTANT"10X,"****"
0816      9 //,40X,"****" IBC=2,H(1)=SINE WAVE
0817      1* ",//,40X,"****" IBC=3,V(1)=CONSTANT
0818      2* ",//,40X,"****" IBC=4,V(1)=SINE WAVE
0819      3* ",//,40X,"****" IBC=5,H(1)=CONSTANT.FINITE CHANNEL
0820      4* ",//,40X,"****" IBC=6,H(1)=SINE WAVE.INFINITE CHANNEL
0821      5* ",//,40X,"****"
0822      6* ",//,40X,"*****"
0823      7**")
0824      20 IF(IPRIN.LT.1) GO TO 45
0825      WRITE(6,1050) T
0826      WRITE(6,1045)
0827      WRITE(6,1055) (I,H(I),V(I),PL(I),PH(I),HC(I),PP(I),I=1,NPOIN)
0828  C*****
0829      IF(IPT.GT.0) GO TO 45
0830      30 WRITE(1,1020)
0831 1020 FORMAT("*****")
0832      1 //,"**** DO YOU WANT TO OUTPUT THE RESULTS ON THE TAPE *****"
0833      2 //,"**** DRIVE? TYPE 1 FOR YES, 0 FOR NO : _")
0834      READ(LUIN,*) ITAPE

```

```

0835      45 IF(ITAPE.LT.1) GO TO 50
0836      WRITE(8,1040) (H(I),V(I),HC(I),PR(I),RL(I),T,I=1,NPOIN)
0837      50 IF(IPT.GT.0) GO TO 57
0838      WRITE(1,1025)
0839      1025 FORMAT("*****")
0840      1      //,"****SPECIFY WHICH OF THE FOLLOWING VARIABLES YOU *****"
0841      2      //,"****WANT PLOTTED : *****"
0842      3      //,"****      0 = NOTHING *****"
0843      4      //,"****      1 = ELEVATION *****"
0844      5      //,"****      2 = VELOCITY *****"
0845      6      //,"****      3 = PRESSURE *****"
0846      7      //,"****      4 = ELEVATION AND VELOCITY *****"
0847      8      //,"****      5 = VELOCITY AND PRESSURE *****"
0848      9      //,"****      6 = ELEVATION AND PRESSURE *****"
0849      1     //,"****      7 = ELEVATION,VELOCITY,AND PRESSURE *****"
0850      2     //,"**** IPLOT = _"
0851      READ(LUIN,*) IPLOT
0852      57 IPT=0
0853      IF(IPLOT.GT.3) GO TO 58
0854      IF(ICPL.LE.0.OR.ICPL.EQ.1) GO TO 59
0855      WRITE(1,1065)
0856      WRITE(1,1030)
0857      1030 FORMAT("*****")
0858      1      //,"**** DO YOU WANT A CONTINUOUS PLOT? *****"
0859      2      //,"**** TYPE 1 FOR YES, 0 FOR NO : _"
0860      READ(LUIN,*) ICPL
0861      ICPL=1
0862      58 IF(IPLOT.LE.0) GO TO 999
0863      GO TO (60,65,70,60,65,60,60), IPLOT
0864      60 CALL XPLOT(H,T,NPOIN,IPT,IPLOT,ICPL)
0865      IPT=IPT+1
0866      IF(IPLOT.EQ.1) GO TO 999
0867      IF(IPLOT.EQ.6) GO TO 70
0868      65 CALL XPLOT(V,T,NPOIN,IPT,IPLOT,ICPL)
0869      IPT=IPT+1
0870      IF(IPLOT.EQ.4) GO TO 999
0871      IF(IPLOT.EQ.2) GO TO 999
0872      70 CALL XPLOT(PR,T,NPOIN,IPT,IPLOT,ICPL)
0873      IPT=IPT+1
0874      999 IF(ICPL.EQ.1) GO TO 100
0875      ICPL=1
0876      C*****
0877      1035 FORMAT("1")
0878      1040 FORMAT(5X,6F12.6)
0879      1045 FORMAT(////, "      NODAL POINT      H-ELEVATION      V-VELOCITY      X
0880      1-COORD.      H-TOTAL      H0-INITIAL      PRESSURE      ",//)
0881      1050 FORMAT("1",//,5X,6HTIME =,F6.3,//)
0882      1055 FORMAT(11F12.6F15.6)
0883      1060 FORMAT(////)
0884      1065 FORMAT("E*dd")
0885      100 RETURN
0886      END
0887      C*****

```

&KYM# T=00003 IS ON CR00011 USING 00168 BLKS R=0000

```

0888      C*****
0889      C*****
0890      SUBROUTINE TOUT(LUIN,NPOIN,IT,IPT), N.C.CHU
0891      C*****
0892      COMMON/BLOC1/ T0,T,TMAX,IFREQ,IBC,NUMBC,IB(2)
0893      COMMON/BLOC3/ H(120),V(120),DH(120),DV(120),PP(120)

```

```

0894      COMMON/BLOC4/ RH(120),RL(120),DX(120),HC(120)
0895      COMMON/BLOC5/ N,W0,DT,GE,H0,XL,CZ,IGEOM,ICZ,AM0
0896      COMMON/BLOC6/ HTN(151),VTN(151),HT(151),VT(151),TIM(151)
0897      C*****
0898      WRITE(1,1040)
0899      WRITE(1,1025) NPOIN,NPLOT
0900      WRITE(1,1030) IT
0901      WRITE(1,1010)
0902      1010 FORMAT("*****")
0903      1      //,"**** DO YOU WANT PRINTER OUTPUT OF THE NUMBER OF      ****"
0904      2      //,"**** POINTS,PLOTS,ETC.? TYPE 1 FOR YES, 0 FOR NO : _")
0905      READ(LUIN,*) JPRIN
0906      25 IF(JPRIN.LT.1) GO TO 30
0907      WRITE(6,1025) NPOIN,NPLOT
0908      WRITE(6,1030) IT
0909      WRITE(6,1035) XL,XL
0910      WRITE(6,1020) (HT(I),VT(I),HTN(I),VTN(I),TIM(I),I=1,IT)
0911      30 WRITE(1,1015)
0912      1015 FORMAT("*****")
0913      1      //,"**** DO YOU WANT TAPE OUTPUT OF ELEVATION AND      ****"
0914      2      //,"**** VELOCITY? TYPE 1 FOR YES, 0 FOR NO : _")
0915      READ(LUIN,*) JTAPE
0916      45 IF(JTAPE.LT.1) GO TO 50
0917      ENDFILE 9
0918      WRITE(8,1020) (HT(I),VT(I),HTN(I),VTN(I),TIM(I),I=1,IT)
0919      ENDFILE 9
0920      50 IPT=0
0921      CALL TPLOT(HTN,TIM,IT,IPT)
0922      IPT=1
0923      CALL TPLOT(VTN,TIM,IT,IPT)
0924      IPT=2
0925      CALL TPLOT(HTI,TIM,IT,IPT)
0926      IPT=1
0927      CALL TPLOT(VTI,TIM,IT,IPT)
0928      C*****
0929      1020 FORMAT(5X,5F16.6)
0930      1025 FORMAT(///,10X,"NUMBER OF POINTS, NP = ",I4,///,
0931      C 10X,"NUMBER OF PLOTS,NPLOT = ",I4)
0932      1030 FORMAT(10X,///," THE # OF POINTS FOR THE TIME PLOT IS = ",I4,///)
0933      1035 FORMAT(13X,"ELEVATION",7X,"VELOCITY",7X,"ELEVATION",8X,"VELOCITY"
0934      1      ,10X,"TIME",//,12X,"AT X=0.00",7X,"AT X=0.00",6X,"AT X=",
0935      2      F5.1,6X,"AT X=",F5.1//)
0936      1040 FORMAT("E=dd")
0937      C*****
0938      RETURN
0939      END

```

*PARTH T=00004 IS ON CR00011 USING 00012 BLKS R=0000

```

0002      C*****
0003      C*****
0004      C*****
0005      SUBROUTINE PARTH(GSM,GF,NP,BV,IB,NBW,NBC), G.A.KERAMIDAS
0006      C*****
0007      C***** SUBROUTINE FOR PARTITIONING OF THE GLOBAL STIFFNESS
0008      C***** MATRIX & FORCE VECTOR BY THE METHOD OF DELETION OF
0009      C***** ROWS & COLUMNS ACCORDING TO THE SPECIFIED
0010      C***** BOUNDARY CONDITIONS.
0011      C***** THE GLOBAL STIFFNESS MATRIX GSM(NP,NBAND) CONTAINS
0012      C***** THE NBAND NON-ZERO DIAGONALS OF THE (NP*NP) MATRIX.
0013      C***** <<WARNING>> PARTH IS USED ONLY FOR (EMA) APPRAYS !
0014      C*****

```

```

0015      DIMENSION BV(NBC), IB(NBC), ND(15)
0016      EMA GSM(NP, NSU), GF(NP, 1)
0017      C*****
0018          N3=(NSU-1)/2
0019          NP4=NP-2*NBC
0020      C*****
0021      DO 10 I=1, NSU
0022          10      ND(I)=I
0023      C*****
0024          ID=0
0025          DO 200 L=1, NBC
0026              ID=ID+1
0027              IP=IB(L)
0028              BC=BV(L)
0029      C*****
0030      C***** MODIFICATION OF THE GLOBAL STIFFNESS MATRIX
0031      C***** & THE FORCE VECTOR
0032      C*****
0033          NS=NSU-N3
0034          NP=IP+NS
0035          IF(NP.GT.NP) GO TO 160
0036          DO 150 I=1, NS
0037              IF(I.EQ.IP) GO TO 150
0038                  JI=IP-I+NS
0039                  GF(I, 1)=GF(I, 1)-GSM(I, JI)*GF(IP, 1)/GSM(IP, NS)
0040          DO 120 J=1, NS
0041              IF(J.EQ.IP) GO TO 120
0042                  JJ=J-IP+NS
0043                  IJ=J-I+NS
0044                  GSM(I, IJ)=GSM(I, IJ)-GSM(I, JI)*GSM(IP, JJ)/GSM(IP, NS)
0045          120      CONTINUE
0046          150      CONTINUE
0047          GO TO 185
0048          160      DO 180 I=1, NS
0049              IN=I+NP4
0050              IF(IN.EQ.IP) GO TO 180
0051                  JL=IP-IN+NS
0052                  GF(IN, 1)=GF(IN, 1)-GSM(IN, JL)*GF(IP, 1)/GSM(IP, NS)
0053          DO 170 J=1, NS
0054              JN=J+NP4
0055              IF(JN.EQ.IP) GO TO 170
0056                  IL=JN-IP+NS
0057                  JC=JN-IN+NS
0058                  GSM(IN, JC)=GSM(IN, JC)-GSM(IN, JL)*GSM(IP, IL)/GSM(IP, NS)
0059          170      CONTINUE
0060          180      CONTINUE
0061          185      GF(IP, 1)=GSM(IP, NS)*BC
0062                  NP0=IP+NS
0063                  IF(NP0.GT.NP) GO TO 195
0064                  DO 190 I0=1, NS
0065                      IF(I0.EQ.IP) GO TO 190
0066                          I1=IP-I0+NS
0067                          I2=I0-IP+NS
0068                          GSM(IP, I2)=0.0
0069                          GSM(I0, I1)=0.0
0070          190      CONTINUE
0071          GO TO 200
0072          195      DO 196 J0=1, NS
0073              JN0=J0+NP4
0074              IF(JN0.EQ.IP) GO TO 196
0075                  J1=IP-JN0+NS
0076                  J2=JN0-IP+NS

```

```

0077      GSK(IP,J2)=0.0
0078      GSK(JN0,J1)=0.0
0079      196  CONTINUE
0080      200  CONTINUE
0081      RETURN
0082      END

```

40COMP T=00004 IS ON CR00011 USING 00008 BLKS R=0000

```

0002 C*****
0003 SUBROUTINE DCOMP(S,INT,A,N,MAXN,M1,M3), A.J.BAKER
0004 C*****
0005 C***** COMPUTE THE LU DECOMPOSITION OF A NONSYMMETRIC
0006 C***** Banded matrix stored in diagonal form by
0007 C***** using partial pivoting.
0008 C*****
0009 DIMENSION S(MAXN,M1),A(MAXN,M3),INT(MAXN)
0010 C*****
0011      L=M1
0012      DO 120 I=1,M1
0013          K2=M1+2-I
0014          DO 100 J=K2,M3
0015              100  A(I,J-L)=A(I,J)
0016              L=L-1
0017              K2=M3-L
0018          DO 110 J=K2,M3
0019              110  A(I,J)=0.0
0020      120  CONTINUE
0021          L=M1
0022          DO 220 K=1,N
0023              X=A(K,1)
0024              I=K
0025              K2=K+1
0026              IF(L.LT.N) L=L+1
0027              IF(L.LT.K2) GO TO 150
0028              DO 140 J=K2,L
0029              IF(ABS(A(J,1))-ABS(X)) 140,140,130
0030              130  X=A(J,1)
0031              I=J
0032      140  CONTINUE
0033      150  INT(K)=I
0034              IF(X) 160,230,160
0035      160  IF(I-K) 170,190,170
0036      170  DO 180 J=1,M3
0037              X=A(K,J)
0038              A(K,J)=A(I,J)
0039      180  A(I,J)=X
0040      190  IF(L.LT.K2) GO TO 220
0041              DO 210 J=K2,L
0042              X=1.0
0043              IF(A(K,1).NE.0.0) X=A(J,1)/A(K,1)
0044              S(K,J-K)=X
0045              DO 200 JJ=2,M3
0046              200  A(J,JJ-1)=A(J,JJ)-A(K,JJ)*X
0047      210  A(J,M3)=0.0
0048      220  CONTINUE
0049      RETURN
0050 C*****
0051 C***** A ZERO PIVOT HAS BEEN FOUND
0052 C***** PRINT ERROR MESSAGE AND STOP
0053 C*****
0054      230  CONTINUE

```

```

0055      WRITE(6,600) K,I,K2,L,J,A(K,I),A(J,I),X
0056 600   FORMAT(//,">>>>> ZERO PIVOT IN DCOMP <<<<<"/,/,
0057      1 10X," K , I , K2 , L , J , A(K,I) , A(J,I) ,
0058      2  X"/,/,10X,5I5,3F10.4)
0059      STOP
0060      END

```

&SOLVE T=00004 IS ON CR00011 USING 00006 BLAS R=0000

```

0002 C*****
0003 C*****
0004 C*****
0005      SUBROUTINE SOLVE(S,INT,F,A,N,MAXN,M1,M3), A.J.BAKER
0006 C*****
0007 C***** COMPUTE THE SOLUTION OF THE LINEAR SYSTEM
0008 C***** USING THE LU DECOMPOSITION OF THE STIFFNESS
0009 C***** MATRIX PROVIDED BY DCOMP AND THE RIGHT
0010 C***** HAND SIDE VECTOR.
0011 C*****
0012      DIMENSION S(MAXN,M1),F(MAXN),A(MAXN,M3),INT(MAXN)
0013 C*****
0014      L=M1
0015      DO 130 K=1,N
0016          I=INT(K)
0017          IF(I-K) 100,110,100
0018      100      X=F(K)
0019              F(K)=F(I)
0020              F(I)=X
0021      110      K2=K+1
0022              IF(L.LT.N) L=L+1
0023              IF(L.LT.K2) GO TO 130
0024              DO 120 I=K2,L
0025                  X=S(K,I-K)
0026      120          F(I)=F(I)-X*F(K)
0027      130      CONTINUE
0028          L=1
0029          DO 180 II=1,N
0030              I=N+1-II
0031              X=F(I)
0032              M=I-1
0033              IF(L-1) 140,160,140
0034      140          DO 150 K=2,L
0035              150      X=X-A(I,K)*F(K+M)
0036      160          F(I)=X/A(I,1)
0037              IF(L-M3) 170,180,180
0038      170          L=L+1
0039      180      CONTINUE
0040      RETURN
0041      END

```