NUREG/CR-2424
PNL-5088-1
Vol. 2

Mathematical Simulation of Sediment and Radionuclide Transport in Coastal Waters

Vol. 2: User's Manual and Computer Program Listing for FETRA

Prepared by Y. Onishi, F. L. Thompson

Pacific Northwest Laboratory

Prepared for
U.S. Nuclear Regulatory Commission

NOTICE

Availability of Reference Materials Cited in NRC Publications

Most documents cited in NRC publications will be available from one of the following sources:

1. The NRC Public Document Room, 1717 H Street, N.W.
   Washington, DC 20555

2. The NRC/GPO Sales Program, U.S. Nuclear Regulatory Commission,
   Washington, DC 20555

3. The National Technical Information Service, Springfield, VA 22161

Although the listing that follows represents the majority of documents cited in NRC publications, it is not intended to be exhaustive.

Referenced documents available for inspection and copying for a fee from the NRC Public Document Room include NRC correspondence and internal NRC memoranda; NRC Office of Inspection and Enforcement bulletins, circulars, information notices, inspection and investigation notices; Licensee Event Reports; vendor reports and correspondence; Commission papers; and applicant and licensee documents and correspondence.

The following documents in the NUREG series are available for purchase from the NRC/GPO Sales Program: formal NRC staff and contractor reports, NRC-sponsored conference proceedings, and NRC booklets and brochures. Also available are Regulatory Guides, NRC regulations in the *Code of Federal Regulations*, and *Nuclear Regulatory Commission Issuances*.

Documents available from the National Technical Information Service include NUREG series reports and technical reports prepared by other federal agencies and reports prepared by the Atomic Energy Commission, forerunner agency to the Nuclear Regulatory Commission.

Documents available from public and special technical libraries include all open literature items, such as books, journal and periodical articles, and transactions. *Federal Register* notices, federal and state legislation, and congressional reports can usually be obtained from these libraries.

Documents such as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings are available for purchase from the organization sponsoring the publication cited.

Single copies of NRC draft reports are available free, to the extent of supply, upon written request to the Division of Technical Information and Document Control, U.S. Nuclear Regulatory Commission, Washington, DC 20555.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at the NRC Library, 7920 Norfolk Avenue, Bethesda, Maryland, and are available there for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

GPO Printed copy price: $4.50

U.S. NUCLEAR REGULATORY COMMISSION DATE: 07/02/84
PACKING SLIP

USNRC
JO ANN RESNER
W 501

ORDER NUMB: N11389-00003
ORDER TYPE: 2

DOCUMENT NO.	QTY @	COST	TOTAL COST	COMMENTS
NUREGCR2424V02	1	.00		

TOTAL DOCUMENTS ORDERED : 1
TOTAL DOCUMENTS ENCLOSED: 1

NUREG/CR-2424
PNL-5088-1
Vol. 2
RE

Mathematical Simulation of Sediment and Radionuclide Transport in Coastal Waters

Vol. 2: User's Manual and Computer Program Listing for FETRA

Manuscript Completed: April 1984
Date Published: May 1984

Prepared by
Y. Onishi, F. L. Thompson

Pacific Northwest Laboratory
Richland, WA 99352

Prepared for
Division of Health, Siting and Waste Management
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
NRC FIN B2271

CONTENTS

1.0 INTRODUCTION

FETRA is a finite element model for simulating the sediment and contaminant transport to surface water (Onishi 1981). The model was applied to a test site in the Irish Sea and modified to account for wave mechanisms that effect sediment suspension (Onishi, Arnold and Mayer 1979). The model consists of three submodels that are coupled to simulate sediment/contaminant interactions. The submodels simulate 1) sediment transport, 2) dissolved contaminant transport, and 3) particulate contaminant (those contaminants absorbed by sediment) transport.

Volume 1 of this report discusses the application of FETRA to the Irish Sea study area to simulate the effect of wind-generated surface waves on the transport of sediment, dissolved ^{137}Cs, and particulate ^{137}Cs. Volume 2 presents a very brief users guide for FETRA in Chapter 2 and a computer program listing of FETRA in Appendix A.

2.0 FETRA MODEL USER'S MANUAL

Most of the data are read by subroutine SPECS. The exceptions include the wind data, read by subroutine RWIND, and time-varying hydrodynamic data, read by subroutine RVEL. FETRA includes an option that allows the use of the coefficient subroutines for the transport code only. This option is considered separately following the main input instructions.

DIMENSIONS

A) Variables used to determine the correct dimensions are as follows:

Bandwidth	= 2*NCOL-1 = largest difference of node numbers within any element
	(See calculations after Card Group 8 in SPECS)
IDN	See Card Group 3
JMAX	See Card Group 20
NBED	See Card Group 18-19
NCNDS	No. of corner nodes - calculated internally when Card Group 9 is read in
NDS	See Card Group 4-5
NE	See Card Group 4-5

B) The dimensions should be assigned as follows:

```
COMMON)BLKI/CT(NDS), C(IDN+2,NDS)
     BLK2/P(NDS, Bandwidth), F(NDS)
     BK3/S(NDS, Bandwidth)
     BLK4/R(NDS), RPAST(NDS,IDN)
     BLK6/VX(NDS), VY(NDS), H(NDS), STRESS(3)
     BLK7/NOD(NE,6), X(NDS), Y(NDS)
     BLK8/MBC(IDN,JMAX), NBC(IDN,JMAX), DB(IDN,JMAX), BC(IDN,JMAX)
          KODE(10)
     BLK9/DX(NE), UY(NE), ALFA(NE), BETA(NE), HS(NE)
     BLK10/PEL(6,6), SEL(6,6), REL(6)
     BLK11/D50(3,NE), BD50(NE), SR(3,NE), SD(3,NE)
     BLK12/ACOF(6), U(6), V(6), D(2), AKJ(9,NE), AKP(3)
```

BLK13/WS(3,NE), CRSTRS(3,NE), CDSTRS(3,NL), ERODA(3,NE)

BLK14/GBA(NE,NBED), GBB(NE,NBED), GBC(Nt,NBD), GBD(NE,NBED)

GBE(NE,NBED), GBF(NE,NBEU), GB3(NE,NBED), POR

BLK15/ILAYR(NE,3), XYSO(NE), BDIV(NE), NBED(NE), BED(NE),
RHOSED(3), XNT(NE,3), RSAVI(NE), RSAV2(NE), RSAV(NE)

BLK17/QLATE(NDS), QSARA(IDN,NE), QPNT(IDN,NDS), CRATE(3,NE)

ARAD(3), ALEFT(3), B2(3), B3(3), GB](3), CS(3), DD(NDS,3)

COMMON/WAVE(NDS), NC(NCNDS), (NCNDS,10), K(NCNDS,10), W(NCNDS,10),
HB(NCNDS), ALPHAC(NCNDS), WAVE,N,D,VIS,DPTHS(NCNDS)

COMMON/WIND/WVEL( ), WANG( ) - dependent on number of wind entries to be used

COMMON/ZTYPE/ZTYP(NCNDS)

Subroutine RVEL — DIMENSION NOD(NCNDS)

Note: DEFINE FILE statements must be consistent with DEFINE FILE in CAFE-I

Subroutine RWAVE

Note: Binary files called internally

Subroutine SAND — NCHECK(NDS)

QCHECK(NDS)

UNITS

Meter, Kilogram (Force), Day, Picocurie

Sediment Concentration: kg/cubic meter of water

Dissolved Chemical: kg or pCi/cubic meter of water

Chemical Attached to Sediment

Input: kg or pCi/cubic meter of water

Output: kg or pCi/kg of sediment

REQUIRED INPUT DATA

CARD GROUP 1-2: Title. Two cards (20A4)

CARD GROUP 3: Identification of Simulation Substances. One card (215)

ID Simulation substance

= 1,2,3 sediment (sand, silt, clay)

= 4 dissolved chemical

= 5,6,7 chemical adsorbed by sediment

= 8 total amount of sediment

= 9 total amount of chemical

(Only one substance is identified. The card groups are repeated for each substance to be simulated.)

IN Flag for coefficient subroutines for transport code

= 0 Only transport subroutines are to be called (See separate input instructions in following section)

/ 0 Used to include sediment/chemical interactions.

CARD GROUP 4-5: Integration Parameters. Two cards. (I) Card I (5I5)

(1) Card 1 (5I5)

IDN Total number of simulation substances

NE Total number of triangular elements

NDS Total number of nodes

NTP Maximum number of time steps to be simulated (0 for steady-state solution)

NPRNT Print frequency (hard copy output for every NPRNT time steps)

(2) Card 2 (2E10.3)

T Time increment (in days)

STYP Numerical solution type

= 0 implies explicit solution

= 0.5 implies Crank-Nicholson solution (default)

= 1 implies implicit solution

CARD GROUP 6-7: Source Information. Two card sets.

(1) Area Source

(a) Card 1 (I5)

NUQSA Number of area sources

(b) Only if NUQSA ≠ 0: Card 2 (2I5, E10.3)

I	Constituent number
J	Source location (element number)
QSARA (I,J)	Source strength (pCi/cubic meter of discharge/day)

(2) Point Source

(a) Card 1 (I5)

NUQSP	Number of point sources

(b) Only if NUQSP ≠ 0 : Card 2 (2I5, E10.3)

I	Constituent number
J	Source location (node number)
QPNT(I,J)	Source strength (pCi/cubic meter of discharge/day)

CARD GROUP 8: Element-Node Connectivity. NE cards (7I5).

I	Element number
NOD (I,K)	Six node numbers corresponding to each element (counterclockwise direction)

CARD GROUP 9: Node Coordinates. NCNDS cards (I5, 2E10.2) where NCNDS is the number of corner nodes.

J	Corner node number (FETRA interpolates to find the the coordinates of midpoints)
X(J)	X-coordinate of j-th node
Y(J)	Y-coordinate of j-th node

(1 Blank Card must follow this card group.)

CARD GROUP 10: Specification of Options. Two cards.

By specifying any of the values given below (in any order on a given card), the associated options are activated. Separate the options with a comma. If no options are desired, input a blank card.

(1) Card 1 [9(A5,1X)]

NUCOF,	Nonuniform coefficients (for card group 15-16-17)	
NUVEL,	For nonuniform velocity and/or depth	
TVB-S,	For time varying boundary conditions	
TVHYD,	For time varying hydraulics	
TVINP,	For time varying input data	
USSDI,	Uniform	Suspended sediment diameter
NSSDI,	Nonuniform	
UBSDI,	Uniform	Bed sediment diameter
NBSDI,	Uniform	
UBGEO,	Uniform	Bed configuration
NBGEO,	Nonuniform	
UBCON,	Uniform	Bed concentration
NBCON,	Nonuniform	
STORE,	Store computed results at the last time step for future restart	
RSTRT,	Restart from the previously stored results	

(2) Card 2 (4A5)

CLAY,	Required if INO (see Card 3) is greater than zero and clay is the limiting bed layer.
SAND,	Required if INO (see Card 3) is greater than zero and sand is the limiting bed layer.
WAVE,	Required if the contaminant and/or sediment transport at any of the nodes is influenced by a wave or surf environment. This will be specified only for sand computation input data stream.
WAVR,	Required if 'WAVE,' is specified and the WAVE and SURF characteristics are to be read from files created by other programs (e.g., LO3D output files).
VELR,	Required if the velocity field is to be read from files created by other programs (e.g., CAFE-I output files).

CARD GROUP 11-12: Sediment Properties. Two card sets.

(1) Only for ID = 1,2,3: 1 Card or NE Cards (5E10.3)

D50(ID,M)	Suspended sediment diameter size
WS(ID,M)	Fall velocity of sediment element M
CRSTRS(ID,M)	Critical shear stress for scouring
CDSTRS(ID,M)	Critical shear stress for deposition
ERODA(ID,M)	Erodability coefficient ID = 1,2,3 at element M

If 'USSDI,' M = 1 (1 card read)

If 'NSSDI,' M = 1, NE (NE cards read)

(2) Only for ID = 1: 1 card or NE cards (E 10.3)

BD50(M) Bed sediment size

If 'UBSDI,' M = 1 (1 card read)

if 'NBSDI,' M = 1, NE (NE cards read)

CARD GROUP 13-14: Initial Conditions. One of the following card sets (A or B):

[A] For uniform initial conditions

(1) Card 1 (A5,1X)

UNICS, Flag for uniform initial conditions

(2) Card 2 (E10.3)

C(ID,1) Initial concentration of substance ID

[B] For nonuniform initial conditions

(1) Card 1 (A5,1X)

NUICS,

(2) NDS Cards (6E10.3)

C(ID,J), J = 1, NDS

[Card Group 13-14 is repeated here for the other (IDN-1) substances].

CARD GROUP 15-16-17: System Properties. Four card sets.

(1) Card 1 (4E10.3)

ALMBDA	Decay rate
AKP(I),I=1,3	Flag for adsorption/desorption = 1 if adsorption/desorption occurs = 0 if no adsorption/desorption

Note: AKP(I) values are no longer read by FETRA

(2) One of the following card sets (a or b):

(a) For <u>nonuniform</u> coefficients (NUCOF option selected): (6E10.3)

[(AKJ(I,J),	Adsorption/desorption constants	
J=1,NE),	I = 1,2,3	Kd value for simulation substance ID = 1,2,3; (m^3/kg)
I=1,9]	I = 4,5,6	Adsorption/desorption rate for simulation substance ID = 1,2,3 in <u>suspended</u> sediment; (1/day)
	I = 7,8,9	Adsorption/desorption rate for simulation substance ID = 1,2,3 in <u>bed</u> sediment; (1/day)

(b) For <u>uniform</u> coefficients (NUCOF <u>not</u> selected): 9 cards (E 10.3)

AKJ(I,I) for I = 1,9 (Defined above)

(3) One Card (7E10.3)

RHOSED(I), I = 1,3	Specific weight of I-th sediment
RHOWAT	Specific weight of water [1000kg (force)/cubic meters]
POR	Porosity
VIS	Kinematic viscosity of water, square meters/day

(4) One Card or NE Cards (5EI0.3)

DX(J)	Dispersion coefficient - X component
DY(J)	Dispersion coefficient - Y component
ALFA(J)	Decay term
BETA(J)	Source/sink term (used only for INO ≠ 0; see following section)
H5(J)	Element thickness

For uniform coefficients (NUCOF option selected): J = 1, NE

For uniform coefficients: J = 1

CARD GROUP 18-19: Bed Layer Inputs. Two card sets.

(1) One Card or NE cards (15, 3E10.3)

NBED(M) Initial number of bed layers

BDIV(M) Thickness of bed layers (used for all layers except top layer).

XYSO(M) Thickness of top bed layer

BED(M) Initial total bed thickness

If 'UBGEO,' M = 1 (1 card read)

If 'NBGEO,' M = 1, NE (NE cards read)

(2) NBED(M) sets of One Card or NE cards (6E10.3)

For element M, bed layer J:

GBA(M,J) = Weight fraction of clay if clay is the limiting bed layer

= Weight fraction of sand if sand is the limiting bed layer

GBB(M,J) = Weight fraction of silt

GBC(M,J) = Weight fraction of sand if clay is the limiting bed layer

= Weight fraction of silt if sand is the limiting bed layer

GBD(M,J) = Chemical concentration of contaminant per unit weight of sediment (clay or sand depending on the sediment type that GBA(M,J) applies to).

GBE(M,J) = Chemical concentration of contaminant per unit weight of silt

GBF(M,J) = Chemical concentration of contaminant per unit weight of sediment (clay or silt depending on the sediment type that GBC(M,J) applies to).

Note: Bed layers are numbered from deepest to shallowest

Note: Appropriate units for GBD, GBE, and BGF are pCi/kg or kg/kg.

If 'UBCOn,' M = 1 (1 card read for each bed layer)

If 'NBCON,' M = 1, NE (NE cards read for each bed layer)

CARD GROUP 20: Boundary Condition Specification. Two card sets.

(1) Card 1 (2I5)

LBC	Number of specified boundary condition nodes
KBC	Number of derivative boundary condition nodes

(2) JMAX Cards [2(I5, E10.2)], where JMAX = larger of LBC, KB,

NBC(ID,J)	Boundary node number
B(ID,J)	Boundary node value (specified)
MBC(ID,J)	Boundary node number
DBC(ID,J)	Boundary node value (derivative)

Note: nodes must be input in a numerically increasing sequence.

CARD GROUP 21: Flow Field Information. One of the following card sets (A, B, or C):

[A] For 'NUVEL,' (nonuniform velocity and/or depth) <u>and</u> 'VELR,' (Velocity field read from files created by other programs)

(1) Card 1 (E10.3)

VFREQ	Frequency of data entries for velocity and/or depth (day)

(2) Card 2 (F10.0) - read by subroutine RVEL

AMAX	Time in days for which velocity and depth files will be repeated

(3) NCNDS Cards (16I5), where NCNDS = no. of corner codes. (Read by Subroutine RVEL)

NOD(I), I=I,NCNDS	FETRA node numbers corresponding to CAFE node numbers (In ascending CAFE node number order)

[B] For 'NUVEL,'where 'VELR,' is <u>not</u> also specified

(1) NDS cards (3EI0.3)

H(J)	Flow depth of j-th node
VX(J)	X-component of velocity at j-th node
VY(J)	Y-component of velocity at j-th node

[C] For a uniform velocity field ('NUVEL,' is not specified)

(1) 1 Card (3E10.3)

H(1)

VX(1) As defined above

VY(1)

CARD'GROUP 22: Wave Sediment Transport Input Data. One of the following card sets.

[A] For 'WAVE,' and 'WAVR,'- subroutine RWAVE reads wave characteristics from files created by another program (e.g., LO3D)

(1) Card 1 (I5, F10.0) - Read by subroutine RWIND

LW	Number of wind data points to be input. The first wind data point will be used at time = 0. days
WFREQ	Time interval in days between each wind data point

(2) (LW/8.) Cards (8E10.2)

WVEL(I)	Wind velocity, m/day
WANG(I)	Direction from which wind is blowing, degrees from true north measured CW

(I=I,LW)

[B] For 'WAVE,' where 'WAVR,' is not also specified - Wave characteristics for each node may be read in or calculated in subroutine WAVSIM

(1) For temporally constant wind: Two cards.

(a) Card ! (A5,IX)

CWIND Flag for constant wind

(b) Card 2 (FI0.0)

WNDVEL Wind velocity (m/day)

(2) For temporally variable wind: Card 1 (A5,1X)

VWIND,

(3)NCNDS card sets, where NCNDS = number of corner nodes

(a) Card 1 (I5,A5,I5)

NODNO Node number

ZTYP (NP) Wave zone type

= WAVE, if node is beyond surf zone and wave characteristics for the node are to be read in

= SURF, if node is in surf zone and surf characteristics for the node are to be read in

= WAVC, if wave characteristics are to be calculated (in subroutine WAVSIM)

NC(NP) Number of wave characteristics to be read for this node (default = I)

(b) If ZTYP = 'WAVE,': NO cards (3E10.4), where NO = NC(NP)

A(NP,I) Wave amplitude, m

K(NP,I) Wave number = 2*π/wave length, 1./m

W(NP,I) Wave freauency = 2*π/wave period, 1./sec

(I=I,NO)

(c) If ZTYP = 'SURF, , (2*NO) cards [2F10.0/(5F10.0)], where NO = NC(NP)

HB(NP) Wave height at breaking, m

ALPHAC(NP) Angle between wave ray and the gradient of the bottom bathymetry, degrees

K(NP,I) Wave number = 2*π/wave length, 1./m

(I=I, NO)

(d)If ZTYP = 'WAVC,': One card (3EI0.4)

(i) For temporally constant wind ('CWIND,'):

DM Mean fetch depth, m

F Effective fetch length, m

(ii) For temporally variable wind ('VWIND,'):

WV Wind velocity, m/day

DM Mean fetch depth, m

F Effective fetch length, m

(4) Blank card

Repeat Card Groups for (ID = 2,IDN):

For (ID = 2,3), the following card groups are required:

- 1-2. Title (The substance being simulated is usually identified on line 2). Two cards (20A4)
- 3. Identification of Simulation Substances. One card (2I5)
- 10. Specification of Options. Two cards
- 11-12 Sediment Properties. First card set only.
- 20. Boundary Condition Specification. Two card sets

For (ID = 4,7), the card groups listed above are required with the exception of card group 11-12.

VFREQ (card group 21) must be repeated at the end of the input data if AMAX (Card Group 21) is less than the amount of time to be simulated. The number of cards (E10.3) to be inserted is equal to (NTP*T/AMAX) (see card group 4-5).

TRANSPORT CODE ONLY

FETRA has an option to run the coefficient subroutines for the transport code only. Sediment/chemical interactions are not accounted for when this option is selected. Card groups required as input for this option are listed below.

Required Input Data

See main input instructions for definitions of parameters. Parameter values specific to the transport code option are noted below.

CARD GROUP 1-2: Title. Two cards (20A4)

CARD GROUP 3: Identification of Simulation Substances. One card (2I5)

ID = 1

INO = 0

CARD GROUP 4-5: Integration Parameters. Two cards (5I5/2E10.3)

IDN = 1

NTP = 0 For steady-state solution only

CARD GROUP 6-7: Source Information. Two blank cards. (Source/sink terms are handled through card group 15-16-17)

CARD GROUP 8: Element-Node Connectivity. NE cards (7I5)

CARD GROUP 9: Node Coordinates. NCNDS cards (I5,2E10.2), where NCNDS is the number of corner nodes.

(1 Blank Card = flag for end of node coordinate data.)

CARD GROUP 10: Specification of Options. Two blank cards

CARD GROUP 13-14: Initial Conditions. (See main input instructions.)

CARD GROUP 15-16-17: System Properties. One card (5E10.3)

DX(I)	Dispersion coefficient - X component
DY(I)	Dispersion coefficient - Y component
ALFA(I)	Decay term
BETA(I)	Source/sink specification used for transport code option
HS(I)	Element thickness

CARD GROUP 20: Boundary Condition Specification. (See main input instructions.)

CARD GROUP 21: Flow Field Information. One card (3E10.3)

H(1)	Flow depth
VX(1)	X-component of velocity
VY(1)	Y-component of velocity

REFERENCES

Onishi Y. 1981. "Sediment-Contaminant Transport Model," Journal of the Hydraulic Division 107(HY9):1089-1107.

Onishi, Y., E. M. Arnold and D. W. Mayer. 1979. Modified Finite Element Transport Model, FETRA, for Sediment and Radionuclides Migration in Open Coastal Waters. NUREG/CR-1026 (PNL-3114). Nuclear Regulatory Commission, Washington, D.C.

APPENDIX

FETRA COMPUTER PROGRAM LISTINGG

```
      PROGRAM FETRA(INPUT,OUTPUT,LUQ,LUH,WAVZON,SURF,TAPE5=INPUT,       FETRA        2
     1 TAPE6=OUTPUT,TAPE7=LUQ,TAPE8=LUH,TAPE3=WAVZON,TAPE4=SURF,        FETFIX1      1
     2 INTAPE,OUTAPE,REPOST,TAPE1=INTAPE,TAPE2=OUTAPE,TAPE9=REPOST)     FETFIX1      2
C      PROGRAM FETRA (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2) FETRA       4
C     FINITE ELEMENT TRANSPORT MODEL                                    FETRA        5
C     THIS PROGRAM SOLVES THE "TIME DEPENDENT" CONVECTION-DIFFUSION     FETRA        6
C     EQUATION IN TWO DIMENSIONS BY THE FINITE ELEMENT METHOD.  THE     FETRA        7
C     GALERKIN PRINCIPLE WAS USED TO FORMULATE THE SYSTEM OF EQUATIONS  FETRA        8
C     FOR A TRIANGULAR ELEMENT.  A QUADRATRIC SHAPE FUNCTION BASED ON   FETRA        9
C     A LAGRANGIAN INTERPOLATING POLYNOMIAL IS USED.  DOMAINS OF COMPLEX FETRA      10
C     GEOMETRY ARE REPRESENTED BY A MESH OF 6-NODE TRIANGULAR ELEMENTS. FETRA       11
C     TIME DEPENDENT MASS OR HEAT TRANSPORT MAY SIMULATED BY DEFINITION FETRA       12
C     OF THE APPROPRIATE SINK AND/OR SOURCE TERMS.                      FETRA       13
C                                                                       FETRA       14
C                                                                       FETRA       15
C     1.  ONISHI,Y.,JOHANSON,P.A.,BACA,R.G.,AND HILTY,E.L.,"STUDIES OF  FETRA       16
C         COLUMBIA RIVER WATER QUALITY---DEVELOPMENT OF MATHEMATICAL    FETRA       17
C         MODELS FOR SEDIMENT AND RADIONUCLIDE TRANSPORT ANALYSIS",     FETRA       18
C         BNWL-B-452, BATTELLE-PACIFIC NORTHWEST LABORATORIES TO THE    FETRA       19
C         U.S.ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION, JAN. 1976 FETRA       20
C     2.  ONISHI,Y.,"FINITE ELEMENT MODELS FOR SEDIMENT AND CONTAMINANT FETRA       21
C         TRANSPORT IN SURFACE WATERS   TRANSPORT OF SEDIMENT AND       FETRA       22
C         RADIONUCLIDES IN THE CLINCH RIVER," BNWL-2227, BATTELLE-      FETRA       23
C         NORTHWEST LABORATORIES, RICHLAND, WASHINGTON,JULY 1977        FETRA       24
C     3   ONISHI,Y.,"MATHEMATICAL SIMULATION OF SEDIMENT AND            FETRA       25
C         RADIONUCLIDE TRANSPORT IN THE COLUMBIA RIVER," BNWL-2228,     FETRA       26
C         BATTELLE-PACIFIC NORTHWEST LABORATORIES, RICHLAND, WASHINGTON FETRA       27
C         AUGUST 1977                                                   FETRA       28
C     4.  SCHECTER,R.S.,"VARIATIONAL METHODS IN ENGINEERING", MCGRAW-   FETRA       29
C         HILL BOOK COMPANY, NEW YORK, 1967.                            FETRA       30
C     5.  DESAI,C.S. AND J.F. ABEL,"INTRODUCTION TO THE FINITE ELEMENT  FETRA       31
C         METHOD", VAN NOSTRAND REIHOLD COMPANY, 1972                   FETRA       32
C     6.  GUYMON,G.L., "APPLICATION OF THE FINITE-ELEMENT METHOD FOR    FETRA       33
C         SIMULATION OF SURFACE WATER TRANSPORT PROBLEMS," INSTITUTE OF FETRA       34
C         WATER RESOURCES, UNIVERSITY OF ALASKA, REPORT NO. IWR-21, 1972 FETRA      35
C     7.  LAPIDUS, L. AND J. SEINFELD,  "NUMERICAL SOLUTION OF ORDINARY FETRA       36
C         DIFFERENTIAL EQUATIONS", ACADEMIC PRESS, INC. 1971.           FETRA       37
C                                                                       FETRA       38
C**** RHOSED OR RHOWAT ARE ACTUALLY SPECIFIC WEIGHTS IN KG(FORCE)/M**3  FETRA       39
C**** RATHER THAN MASS DENSITY. THESE SHOULD BE CHANGED TO SPWSED AND   FETRA       40
C**** SPWWAT.                                                           FETRA       41
C                                                                       FETRA       42
      COMMON NDS                                                        FETRA       43
      COMMON /BLK1/CT(240),C(9,240)                                     FETRA       44
      COMMON /BLK2/P(240,86),F(240)                                     FETRA       45
      LEVEL  2,P,F                                                      FETFIX4      1
      COMMON /BLK3/S(240,86)                                            FETRA       46
      LEVEL  2,S                                                        FETFIX4      2
      COMMON /BLK4/ R(240), RPAST(240,7), NODBET, BETA1, AREA1          FETRA       47
      LEVEL  2,R,RPAST,NODBET,BETA1,AREA1                               FETFIX4      3
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)                     FETRA       48
      COMMON /BLK7/NOD(240,6),X(240),Y(240)                             FETRA       49
      COMMON /BLK8/KBC,LBC,MBC(7,120),NBC(7,120),DBC(7,120),BC(7,120),  FETRA       50
     1 KODE(10)                                                         FETRA       51
      COMMON /BLK11/DS0(3,100),BD50(100),SR(3,100),SD(3,100)            FETRA       52
      COMMON /BLK12/ACOF(6),U(6),V(6),D(2),AKJ(9,100),ALMBDA,RHOWAT,    FETRA       53
     1             AKP(3)                                               FETRA       54
      COMMON /BLK13/WS(3,100),CRSTRS(3,100),CDSTRS(3,100),ERODA(3,100)  FETRA       55
      COMMON /BLK14/GBA(100,10),GBB(100,10),GBC(100,10),GBD(100,10),    FETRA       56
     1             GBE(100,10),GBF(100,10),GBG(100,10),POR              FETRA       57
      COMMON /BLK15/ILAYR(100,3),XYS0(100),BDIV(100),NBED(100),BED(100), FETRA      58
     1             RHOSED(3),XNT(100,3),RSAV1(100),RSAV2(100),RSAV3(100)  FETRA     59
```

```
      COMMON /BLK17/QLATE(240),QSARA(7,100),QPNT(7,240),CRATE(3,100),   FETRA      60
     1  CD(4,240)                                                       FETRA      61
      LEVEL   2,QLATE,QSARA,QPNT,CRATE,CD                               FETFIX4     4
      LOGICAL RSTRT,STORE                                               FETRA      62
      DIMENSION ARAD(3),ALEFT(3),B2(3),B3(3),GBJ(3),CS(3)               FETRA      63
C                                                                       FETRA      64
C                                                                       FETRA      65
      DIMENSION DD(240,3)                                               FETRA      66
      DATA EPSI/1.0E-12/                                                FETRA      67
C                                                                       FETRA      68
C                                                                       FETRA      69
C                                                                       FETRA      70
      K=1                                                               FETRA      71
      KKK=1                                                             FETRA      72
      TSPEC = 0.0                                                       FETRA      73
      TTRANS = 0.0                                                      FETRA      74
      TBNDRY = 0.0                                                      FETRA      75
      TAMSOL = 0.0                                                      FETRA      76
      AT=0.0                                                            FETRA      77
      RAT=0.0                                                           FETRA      78
C                                                                       FETRA      79
C     INPUT THE PROBLEM SPECIFICATIONS                                  FETRA      80
C                                                                       FETRA      81
C     COMPUTATIONAL LOOP                                                FETRA      82
C                                                                       FETRA      83
100   CONTINUE                                                          FETRA      84
      T1 = SECOND(0.0)                                                  FETFIX1     3
      CALL SPECS(NE,NDS,NCOL,NTP,T,ID,IDN,INO,K,NPRNT,RSTRT,STORE,KKK,  FETRA      86
     1 RAT)                                                             FETRA      87
      TT1 = SECOND(T1)                                                  FETFIX1     4
      TSPEC = TSPEC + TT1                                               FETRA      89
      IF(K .NE. 1) GO TO 120                                            FETRA      90
      DO 115 I=1,NDS                                                    FETRA      91
  115 RPAST(I,ID) = 0.0                                                 FETRA      92
  120 NROW=NDS                                                          FETRA      93
       IF(ID.GT.4) GO TO 126                                            FETRA      94
       DO 125 I=1,NDS                                                   FETRA      95
       CD(ID,I) = C(ID,I)                                               FETRA      96
       IF(ID.LE.3.AND.CD(ID,I).LT.0.0) CD(ID,I) = 0.0                   FETRA      97
 125   CONTINUE                                                         FETRA      98
 126   CONTINUE                                                         FETRA      99
C                                                                       FETRA     100
C     FORM THE COEFFICIENT MATRICES +                                   FETRA     101
C                     (P),    (S),    (R)                               FETRA     102
C                                                                       FETRA     103
      T2 = SECOND(0.0)                                                  FETFIX1     5
C     H3 SPECIAL                                                        FETRA     105
C     IF(ID.NE.4) GO TO 997                                             FETRA     106
      CALL TRANSP(NE,NROW,NCOL,T,ID,IDN,INO,K,NPRNT,NTP)                FETRA     107
      TT2 = SECOND(T2)                                                  FETFIX1     6
      TTRANS = TTRANS + TT2                                             FETRA     109
C                                                                       FETRA     110
C     SET THE SPECIFIED BOUNDARY CONDITIONS IN THE LOAD VECTOR (R)      FETRA     111
C                                                                       FETRA     112
      T3 = SECOND(0.0)                                                  FETFIX1     7
      CALL BNDRYS (NROW,NCOL,ID)                                        FETRA     114
      TT3 = SECOND(T3)                                                  FETFIX1     8
      TBNDRY = TBNDRY + TT3                                             FETRA     116
C                                                                       FETRA     117
      IF (NTP.GT.0 ) GO TO 130                                          FETRA     118
C     COMPUTE STEADY STATE SOLUTION FOR TWO-DIMENSIONAL CONVECTION-     FETRA     119
C     DIFFUTION EQUATION                                                FETRA     120
```

```
      WRITE (6,270)                                                     FETRA      121
      CALL BANSOL (NROW,NCOL,S,R)                                       FETRA      122
      DO 140 I=1,NROW                                                   FETRA      123
      CT(I)=R(I)                                                        FETRA      124
  140 CONTINUE                                                          FETRA      125
      GO TO 160                                                         FETRA      126
130   CONTINUE                                                          FETRA      127
C                                                                       FETRA      128
C     COMPUTE THE TIME DEPENDENT SOLUTION IF REQUIRED                   FETRA      129
C                                                                       FETRA      130
C     H3 SPECIAL  CHANGE 1 TO 4                                         FETRA      131
      IF(KKK.EQ.1.AND.ID.EQ.1) AT=AT+RAT                                FETRA      132
      IF (ID.EQ.1) AT=AT+T                                              FETRA      133
      IF(MOD(K,NPRNT) .NE. 0 .AND. K .NE. NTP) GO TO 150                FETRA      134
      WRITE (6,280)                                                     FETRA      135
      WRITE (6,225) K,T,AT,ID                                           FETRA      136
150   CONTINUE                                                          FETRA      137
      T4 = SECOND(0.0)                                                  FETFIX1      9
      CALL AMSOL(NROW,NCOL,ID,T)                                        FETRA      139
      TT4 = SECOND(T4)                                                  FETFIX1     10
      TAMSOL = TAMSOL + TT4                                             FETRA      141
      DO 155 I=1,NROW                                                   FETRA      142
  155 RPAST(I,ID) = R(I)                                                FETRA      143
C                                                                       FETRA      144
C     OUTPUT THE RESULTS TO THE PRINTER                                 FETRA      145
C                                                                       FETRA      146
160   CONTINUE                                                          FETRA      147
      N=0                                                               FETRA      148
      J=0                                                               FETRA      149
      DO 170 I=1,LBC                                                    FETRA      150
      II=NBC(ID,I)                                                      FETRA      151
170   C(ID,II)=BC(ID,I)                                                 FETRA      152
180   N=N+1                                                             FETRA      153
      J=J+1                                                             FETRA      154
      DO 190 I=1,LBC                                                    FETRA      155
      IF (NBC(ID,I).EQ.J) J=J+1                                         FETRA      156
190   CONTINUE                                                          FETRA      157
      IF (J.GT.NOS) GO TO 200                                           FETRA      158
      C(ID,J)=CT(N)                                                     FETRA      159
      IF (N.LT.NROW) GO TO 180                                          FETRA      160
200   CONTINUE                                                          FETRA      161
      IF(MOD(K,NPRNT) .NE. 0 .AND. K .NE. NTP) GO TO 212                FETRA      162
C                                                                       FETRA      163
      IF (ID.LT.5) GO TO 202                                            FETRA      164
      DO 201 J=1,NOS                                                    FETRA      165
      IF (C(ID-4,J).GT.1.E-6)                                           FETRA      166
     *   C(ID,J) = C(ID,J)/C(ID-4,J)                                    FETRA      167
      IF (C(ID-4,J).LE.1.E-6)                                           FETRA      168
     *   DO(J,ID-4)=C(ID,J)                                             FETRA      169
      IF (C(ID-4,J).LE.1.E-6)                                           FETRA      170
     *   C(ID,J) = 0.0                                                  FETRA      171
  201 CONTINUE                                                          FETRA      172
  202 CONTINUE                                                          FETRA      173
      WRITE (6,230)                                                     FETRA      174
      WRITE(6,240) ID                                                   FETRA      175
      NPRT=NOS/6+1                                                      FETRA      176
      DO 210 I=1,NPRT                                                   FETRA      177
      NST=(I-1)*6+1                                                     FETRA      178
      IF (NST.GT.NOS) GO TO 210                                         FETRA      179
      II=NST+5                                                          FETRA      180
      IF (II.GT.NOS) II=NOS                                             FETRA      181
      WRITE (6,310) (J,C(ID,J),J=NST,II)                                FETRA      182
```

```
210   CONTINUE
      WRITE (9) (J,C(ID,J),J=1,NDS)
C
      IF (ID.LT.5) GO TO 212
      DO 203 J = 1,NDS
      IF (C(ID-4,J).GT.1.E-6)
     #   C(ID,J) = C(ID,J)*C(ID-4,J)
      IF (C(ID-4,J).LE.1.E-6)
     #   C(ID,J) = DD(J,ID-4)
203   CONTINUE
  212 CONTINUE
      IF (INO.LE.0) GO TO 380
C     IF (ID.EQ.IDN) CALL BEDHIS (NE,T)
C     H3 SPECIAL
 997  CONTINUE
      ID=ID+1
      IF ((ID-IDN).LE.0) GO TO 100
      ID=1
C     H3 SPECIAL
C     IF(ID.GT.0) GO TO 998
      IF(MOD(K,NPRNT) .NE. 0 .AND. K .NE. NTP) GO TO 380
C
C**   CALCULATE AND OUTPUT TOTAL SUSPENDED SEDIMENT
C**   CONCENTRATION, KG/M**3 OF WATER.
C
      DO 104 J=1,NDS
      C(8,J)=0.
      DO 104 I=1,3
  104 C(8,J)=C(8,J)+C(I,J)
      WRITE(6,290)
      WRITE(6,320)
      WRITE(6,310) (J,C(8,J),J=1,NDS)
      WRITE (9) (J,C(8,J),J=1,NDS)
C
C**   CALCULATE AND OUTPUT THE TOTAL CONCENTRATION OF CONTAMINANT
C**   ATTACHED TO SUSPENDED SEDIMENT, PCI/M**3 OF WATER.
C
      DO 105 J=1,NDS
      C(9,J)=0.
      DO 105 I=1,3
  105 C(9,J)=C(9,J)+C(I+4,J)
      WRITE(6,235)
      WRITE(6,325)
      WRITE(6,310) (J,C(9,J),J=1,NDS)
      WRITE (9) (J,C(9,J),J=1,NDS)
C
C**   CALCULATE AND OUTPUT THE TOTAL SUSPENDED AND
C**   DISSOLVED CONTAMINANT CONCENTRATION, PCI/M**3 OF WATER.
C
      DO 106 J=1,NDS
  106 C(9,J)=C(9,J)+C(4,J)
      WRITE(6,235)
      WRITE(6,326)
      WRITE(6,310) (J,C(9,J),J=1,NDS)
      WRITE (9) (J,C(9,J),J=1,NDS)
C      H3 SPECIAL
C 998  CONTINUE
C      THAS[illegible](-0.25*C(4,160)-0.25*C(4,186)+0.5*C(4,188)+
C    1 2.0*C(4,173)+
C    2 3.0*C(4,187)+3.0*C(4,174)-0.5*C(4,160)+0.26*C(4,188)+
C    3 0.24*C(4,162)+4.0*C(4,174)+5.0*C(4,175)+3.96*C(4,161)-
C    4 0.186*C(4,162)-0.103*C(4,188)+0.348*C(4,164)+6.66*C(4,175)+
```

```
C     5 7.37*C(4,176)+7.34*C(4,163)+0.209*C(4,164)-0.302*C(4,188)+          FETRA     245
C     6 0.093*C(4,190)+7.93*C(4,176)+7.77*C(4,186)+8.46*C(4,177))           FETRA     246
C        THASS=THASS+(- 0.25*C(4,186)-0.252*C(4,212)+                       FETRA     247
C     1 0.504*C(4,188)+2.01*C(4,199)+                                       FETRA     248
C     2 3.02*C(4,200)+3.02*C(4,167)+0.240*C(4,188)-0.516*C(4,212)+          FETRA     249
C     3 0.259*C(4,214)+4.07*C(4,200)+4.12*C(4,213)+5.13*C(4,201)-           FETRA     250
C     4 0.142*C(4,188)-0.106*C(4,214)+0.248*C(4,216)+6.65*C(4,201)+         FETRA     251
C     5 7.17*C(4,215)+7.13*C(4,202)-0.261*C(4,188)+0.128*C(4,216)+          FETRA     252
C     6 0.134*C(4,190)+7.60*C(4,202)+8.13*C(4,203)+7.61*C(4,189))           FETRA     253
C        THASS = 2.0E6 * THASS                                              FETRA     254
C        WRITE (6,780) K,THASS                                              FETRA     255
C780  FORMAT (" K, THASS = ",I5,1PE15.4)                                     FETRA     256
C                                                                           FETRA     257
      THASS = 0.0                                                           FETRA     258
      PAMASS = 0.0                                                          FETFIX6     1
      TOTBED = 0.0                                                          FETFIX6     2
      DO 800 M=1,NE                                                         FETRA     259
      N1 = NOD(M,1)                                                         FETRA     260
      N2 = NOD(M,2)                                                         FETRA     261
      N3 = NOD(M,3)                                                         FETRA     262
      N4 = NOD(M,4)                                                         FETRA     263
      N5 = NOD(M,5)                                                         FETRA     264
      N6 = NOD(M,6)                                                         FETRA     265
      A2 = X(N1)-X(N3)                                                      FETRA     266
      A3 = X(N2)-X(N1)                                                      FETRA     267
      A5 = Y(N3)-Y(N1)                                                      FETRA     268
      A6 = Y(N1)-Y(N2)                                                      FETRA     269
      ARE = (A3*A5-A2*A6)/2.0                                               FETRA     270
      THASS = THASS + ((H(N1)/30.-H(N2)/60.-H(N3)/60.)*C(4,N1) +            FETRA     271
     1 (-H(N1)/60.+H(N2)/30.-H(N3)/60.)*C(4,N2) + (-H(N1)/60.-              FETRA     272
     2 H(N2)/60.+H(N3)/30.)*C(4,N3) + (H(N1)/7.5+H(N2)/7.5+                 FETRA     273
     3 H(N3)/15.)*C(4,N4) + (H(N1)/15.+H(N2)/7.5+H(N3)/7.5)*                FETRA     274
     4 C(4,N5) + (H(N1)/7.5+H(N2)/15.+H(N3)/7.5)*C(4,N6))*ARE               FETRA     275
C                                                                           FETFIX6     3
      PAMASS = PAMASS + ((H(N1)/30.-H(N2)/60.-H(N3)/60.)*C(9,N1) +          FETFIX6     4
     1 (-H(N1)/60.+H(N2)/30.-H(N3)/60.)*C(9,N2) + (-H(N1)/60.-              FETFIX6     5
     2 H(N2)/60.+H(N3)/30.)*C(9,N3) + (H(N1)/7.5+H(N2)/7.5+                 FETFIX6     6
     3 H(N3)/15.)*C(9,N4) + (H(N1)/15.+H(N2)/7.5+H(N3)/7.5)*                FETFIX6     7
     4 C(9,N5) + (H(N1)/7.5+H(N2)/15.+H(N3)/7.5)*C(9,N6))*ARE               FETFIX6     8
C                                                                           FETFIX6     9
      NBE1 = NBED(M)                                                        FETFIX6    10
      TOTAL = GBA(M,NBE1)/RHOSED(1)+GBB(M,NBE1)/RHOSED(2)                   FETFIX6    11
     1 +GBC(M,NBE1)/RHOSED(3)                                               FETFIX6    12
      XNO = (1.0-POR)/TOTAL                                                 FETFIX6    13
      TOTBED = TOTBED+GBG(M,NBE1)*XND*XYSO(M)*ARE                           FETFIX6    14
      NBED1 = NBED(M)-1                                                     FETFIX6    15
      DO 800 J=1,NBED1                                                      FETFIX6    16
      TOTAL = GBA(M,J)/RHOSED(1)+GBB(M,J)/RHOSED(2)                         FETFIX6    17
     1 +GBC(M,J)/RHOSED(3)                                                  FETFIX6    18
      XNO = (1.0-POR)/TOTAL                                                 FETFIX6    19
      TOTBED = TOTBED + GBG(M,J)*XND*BDIV(M)*ARE                            FETFIX6    20
 800  CONTINUE                                                              FETRA     276
      PAMASS = PAMASS-THASS                                                 FETFIX6    21
      TOTMAS = THASS+PAMASS+TOTBED                                          FETFIX6    22
      WRITE(6,702) TOTMAS,THASS,PAMASS,TOTBED                               FETFIX6    23
702   FORMAT (/"TOTAL MASS = ",1PE15.4,2X,"DISSOLVED MASS = ",1PE15.4,      FETFIX6    24
     1 2X,"PARTICULATE MASS = ",1PE15.4,2X,"BED MASS = ",1PE15.4/)          FETFIX6    25
C                                                                           FETRA     279
C     H3 SPECIAL                                                            FETRA     280
C     IF(ID.GT.0) GO TO 999                                                 FETRA     281
C**   CALCULATE AND OUTPUT THE WEIGHTED AVERAGE OF                          FETRA     282
C**   CONTAMINANT ATTACHED TO SUSPENDED SEDIMENT,                           FETRA     283
```

```
C**   PCI/KG OF SEDIMENT.
C
      DO 107 J=1,NDS
      IF(C(8,J) .EQ. 0.) GO TO 107
      C(9,J)=(C(9,J)-C(4,J))/C(8,J)
  107 CONTINUE
      WRITE(6,235)
      WRITE(6,330)
      WRITE(6,310) (J,C(9,J),J=1,NDS)
      WRITE (9) (J,C(9,J),J=1,NDS)
      WRITE(6,290)
      WRITE (6,600)
      DO 370 I=1,NE
      NBE1=NBED(I)
      WRITE(6,601) I,BED(I),GBA(I,NBE1),GBB(I,NBE1),GBC(I,NBE1),GBD(I,
     1NBE1),GBE(I,NBE1),GBF(I,NBE1),GBG(I,NBE1)
      XX=BED(I)-XYS0(I)
      NBED1=NBED(I)-1
      DO 370 LM=1,NBED1
      XY=XX-BDIV(I)*FLOAT(LM-1)
      LN=NBED(I)-LM
      WRITE(6,602)XY,GBA(I,LN),GBB(I,LN),GBC(I,LN),GBD(I,LN),GBE(I,LN),
     1GBF(I,LN),GBG(I,LN)
370   CONTINUE
       WRITE(9) (I,BED(I),GBA(I,4),GBB(I,4),GBC(I,4),GBD(I,4),
     1    GBE(I,4),GBF(I,4),GBG(I,4),I=1,NE)
380   CONTINUE
C      WRITE (6,700) ID,K,TSPEC,TTRANS,TBNDRY,TAMSOL
C700   FORMAT (" TOTAL TIME - ID,K,SPEC,TRANS,BNDRYS,AMSOL ="
C     1 2I5,4F12.4)
C     H3 SPECIL
C 999  CONTINUE
      IF (K.GE.NTP) GO TO 350
      K=K+1
      KKK=KKK+1
      GO TO 100
350   CONTINUE
      K = K+1
      IF(STORE.AND.K.GE.NTP) WRITE(2) AT,K,C,NBED,XYS0,BED,GBA,GBB,GBC,
     1 GBD,GBE,GBF,GBG
      STOP
225   FORMAT (9X,"TIME SEGMENT NO. ",I5,/,9X,"TIME STEP SIZE  ",4X,1PE11
     1.4,/,9X,"COMPUTED TIME PLANE",1X,1PE11.4,/,9X,"SIMU. SUBSTANCE ID=
     2",3X,I5,//)
230   FORMAT (//,10X,"LIST THE RESULTS FOR EACH NODE",//)
  235 FORMAT(//)
  240 FORMAT(10X,"NODE",5X,"CONCENTRATION OF SUBSTANCE ID=",I3,/)
270   FORMAT (1H1,//,10X,"STEADY STATE SOLUTION",//)
280   FORMAT (1H1,//,10X,"TIME DEPENDENT SOLUTION",//)
290   FORMAT (1H1)
  310 FORMAT((10X,6(I5,1PE15.7)))
  320 FORMAT(10X,"NODE",5X,"TOTAL SEDIMENT CONCENTRATION, KGF/M**3",/)
  325 FORMAT(10X,"NODE",5X,"TOTAL PARTICULATE CONTAMINANT CONCENTRATION
     $ATTACHED TO SEDIMENT",/)
  326 FORMAT(10X,"NODE",5X,"TOTAL PARTICULATE AND DISSOLVED CONTAMINANT
     $CONCENTRATION",/)
  330 FORMAT(10X,"NODE",5X,"WEIGHTED AVERAGE PARTICULATE CONTAMINANT CON
     $CENTRATION ATTACHED TO SEDIMENT",/)
 600  FORMAT(9X,"I",7X,"BED",12X,"GBA",12X,"GBB",12X,"GBC",12X,"GBD",12
     1X,"GBE",12X,"GBF",12X,"GBG",/)
 601  FORMAT(5X,I5,/,10X,8(1PE15.7))
 602  FORMAT(10X,8(1PE15.7))
```

```
      END                                                               FETRA     346
      SUBROUTINE ACOEFS(M,AREA)                                         ACOEFS      2
C     THIS SUBROUTINE COMPUTES THE ELEMENT COEFFICIENTS FORMING THE     ACOEFS      3
C     MATRICES OF THE M-TH ELEMENT.                                     ACOEFS      4
C                                                                       ACOEFS      5
      COMMON /BLK7/NOD(240,6),X(240),Y(240)                             ACOEFS      6
      COMMON /BLK12/ACOF(6),U(6),V(6),D(2),AKJ(9,100),ALMBDA,RHOMAT,    ACOEFS      7
     1              AKP(3)                                              ACOEFS      8
C                                                                       ACOEFS      9
      N1=NOD(M,1)                                                       ACOEFS     10
      N2=NOD(M,2)                                                       ACOEFS     11
      N3=NOD(M,3)                                                       ACOEFS     12
C                                                                       ACOEFS     13
      ACOF(1)=X(N3)-X(N2)                                               ACOEFS     14
      ACOF(2)=X(N1)-X(N3)                                               ACOEFS     15
      ACOF(3)=X(N2)-X(N1)                                               ACOEFS     16
      ACOF(4)=Y(N2)-Y(N3)                                               ACOEFS     17
      ACOF(5)=Y(N3)-Y(N1)                                               ACOEFS     18
      ACOF(6)=Y(N1)-Y(N2)                                               ACOEFS     19
C                                                                       ACOEFS     20
C     COMPUTE THE AREA OF M-TH TRIANGLE FROM THE LOCAL COORDINATES.     ACOEFS     21
C                                                                       ACOEFS     22
      AREA=(ACOF(3)*ACOF(5)-ACOF(2)*ACOF(6))/2.                         ACOEFS     23
C                                                                       ACOEFS     24
      IF (AREA.GT.0.) GO TO 100                                         ACOEFS     25
      WRITE (6,110) AREA,M                                              ACOEFS     26
      WRITE (6,115) N1,N2,N3                                            ACOEFS     27
115   FORMAT (" N1,N2,N3 ",3I5)                                         ACOEFS     28
      WRITE (6,120) X(N3),X(N2)                                         ACOEFS     29
      WRITE (6,120) X(N1),X(N3)                                         ACOEFS     30
      WRITE (6,120) X(N2),X(N1)                                         ACOEFS     31
      WRITE (6,125) Y(N2),Y(N3)                                         ACOEFS     32
      WRITE (6,125) Y(N3),Y(N1)                                         ACOEFS     33
      WRITE (6,125) Y(N1),Y(N2)                                         ACOEFS     34
120   FORMAT (" X ",1P2E14.4)                                           ACOEFS     35
125   FORMAT (" Y ",1P2E14.4)                                           ACOEFS     36
      WRITE (6,130) ACOF(3),ACOF(5),ACOF(2),ACOF(6)                     ACOEFS     37
130   FORMAT (" ACOF 3,5,2,6 ",1P4E14.4)                                ACOEFS     38
      STOP                                                              ACOEFS     39
C                                                                       ACOEFS     40
100   CONTINUE                                                          ACOEFS     41
      RETURN                                                            ACOEFS     42
110   FORMAT (//,10X,"NEGATIVE OR ZERO AREA ",1PE12.4," ELEMENT",I5)    ACOEFS     43
      END                                                               ACOEFS     44
      SUBROUTINE AMSOL(NROW,NCOL,ID,T)                                  AMSOL       2
C                                                                       AMSOL       3
C**   THIS SUBROUTINE USES A NUMERICAL APPROXIMATION TO                 AMSOL       4
C**   SOLVE THE SYSTEM OF ORDINARY DIFFERENTIAL EQUATIONS.              AMSOL       5
C**   STYP=0. IMPLIES AN EXPLICIT SOLUTION                              AMSOL       6
C**   STYP=.5 IMPLIES A CRANK-NICHOLSON SOLUTION                        AMSOL       7
C**   STYP=1. IMPLIES AN IMPLICIT SOLUTION                              AMSOL       8
C                                                                       AMSOL       9
      COMMON /BLK1/CT(240),C(9,240)                                     AMSOL      10
      COMMON /BLK2/P(240,86),F(240)                                     AMSOL      11
      LEVEL  2,P,F                                                      FETFIX4     5
      COMMON /BLK3/S(240,86)                                            AMSOL      12
      LEVEL  2,S                                                        FETFIX4     6
      COMMON /BLK4/ R(240), RPAST(240,7), NODBET, BETA1, AREA1          AMSOL      13
      LEVEL  2,R,RPAST,NODBET,BETA1,AREA1                               FETFIX4     7
      COMMON /BLK16/ STYP                                               AMSOL      14
      COMMON /BLK17/QLATE(240),QSARA(7,100),QPNT(7,240),CRATE(3,100),   AMSOL      15
     1  CD(4,240)                                                       AMSOL      16
```

```
      LEVEL    2,QLATE,QSARA,QPNT,CRATE,CO                                 FETFIX4
C                                                                           AMSOL
      NBND=2*NCOL-1                                                         AMSOL
      DO 120 J=1,NBND                                                       AMSOL
      DO 120 I=1,NROW                                                       AMSOL
      PBAR=P(I,J)+T*S(I,J)*STYP                                             AMSOL
      SBAR=P(I,J)-T*S(I,J)*(1.-STYP)                                        AMSOL
      P(I,J)=PBAR                                                           AMSOL
      S(I,J)=SBAR                                                           AMSOL
  120 CONTINUE                                                              AMSOL
C                                                                           AMSOL
C      IF (ID.LT.5)  GO TO 302                                              AMSOL
C      WRITE(6,399)                                                         AMSOL
C      WRITE(6,400) (R(J),J=1,NROW)                                         AMSOL
C      WRITE(6,401)                                                         AMSOL
C      DO 300 I=1,NROW                                                      AMSOL
C      WRITE(6,400) (S(I,J),J=1,NBND)                                       AMSOL
C 300  CONTINUE                                                             AMSOL
C      WRITE(6,402)                                                         AMSOL
C      DO 301 I=1,NROW                                                      AMSOL
C      WRITE(6,400) (P(I,J),J=1,NBND)                                       AMSOL
C 301  CONTINUE                                                             AMSOL
C 302  CONTINUE                                                             AMSOL
C                                                                           AMSOL
C                                                                           AMSOL
      CALL COMB(NROW,NCOL,ID)                                               AMSOL
      DO 130 I=1,NROW                                                       AMSOL
  130 F(I)=F(I)+T*R(I)*STYP+T*RPAST(I,ID)*(1.-STYP)                         AMSOL
C                                                                           AMSOL
C                                                                           AMSOL
C      WRITE(6,406)                                                         AMSOL
C 406  FORMAT(//,10X,"R MATRIX",/)                                          AMSOL
C      WRITE(6,405)  (R(I),I=1,NROW)                                        AMSOL
C      WRITE(6,407)                                                         AMSOL
C 407  FORMAT(//,10X,"RPAST MATRIX",/)                                      AMSOL
C      WRITE(6,405)  (RPAST(I,ID),I=1,NROW)                                 AMSOL
C                                                                           AMSOL
C                                                                           AMSOL
C      IF (ID.LT.5) GO TO 307                                               AMSOL
C      WRITE(6,403)                                                         AMSOL
C      DO 999 I=1,NROW                                                      AMSOL
C      WRITE(6,405)  (P(I,J),J=1,NBND)                                      AMSOL
C999   CONTINUE                                                             AMSOL
C      WRITE(6,404)                                                         AMSOL
C      WRITE(6,405)  (F(J),J=1,NROW)                                        AMSOL
C 307  CONTINUE                                                             AMSOL
C                                                                           AMSOL
C                                                                           AMSOL
      CALL BANSOL(NROW,NCOL,P,F)                                            AMSOL
      DO 140 I=1,NROW                                                       AMSOL
      CT(I)=F(I)                                                            AMSOL
  140 CONTINUE                                                              AMSOL
      RETURN                                                                AMSOL
  399 FORMAT(//,10X,"R MATRIX",/)                                           AMSOL
  400 FORMAT(12(1PE10.3))                                                   AMSOL
  401 FORMAT(//,10X,"S MATRIX",/)                                           AMSOL
  402 FORMAT(//,10X,"P MATRIX",/)                                           AMSOL
  403 FORMAT(//,10X,"P MATRIX FOR PX = F",/)                                AMSOL
  404 FORMAT(//,10X,"F MATRIX FOR PX = F",/)                                AMSOL
  405 FORMAT(12(1PE10.3))                                                   AMSOL
      END                                                                   AMSOL
       SUBROUTINE BANSOL(NROW,NCOL,P,F)                                     BANSOL
```

```
C
C     THIS ROUTINE SOLVES THE LINEAR SYSTEM OF EQUATIONS (P) X = (F)
C     WHERE (P) IS BANDED AND UNSYMMETRIC.  THE ALGORITHM EMPLOYED IS
C     GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING.
C
      LEVEL  2,P,F
      DIMENSION P(240,86),F(240)
      DATA EPSI/1.E-36/
C
      N=NROW
      L=NCOL
      M=2*NCOL-1
      N1=N-1
C
C     FORWARD ELIMINATION WITH PARTIAL PIVOTING.
C
      DO 160 I=1,N1
      IPIV=I
      IRE=I+1
      DO 100 IR=IRE,L
      IF (ABS(P(IR,1)) .LE. ABS(P(I,1))) GO TO 100
      IPIV=IR
  100 CONTINUE
C
      IF (IPIV .EQ. I) GO TO 120
      T=F(I)
      F(I)=F(IPIV)
      F(IPIV)=T
      DO 110 J=1,M
      T=P(I,J)
      P(I,J)=P(IPIV,J)
  110 P(IPIV,J)=T
  120 CONTINUE
      IF(ABS(P(I,1)) .LE. EPSI) GO TO 135
      F(I)=F(I)/P(I,1)
      DO 130 J=2,M
      P(I,J)=P(I,J)/P(I,1)
  130 CONTINUE
  135 CONTINUE
      DO 150 IR=IRE,L
      T=P(IR,1)
      F(IR)=F(IR)-T*F(I)
      DO 140 J=2,M
  140 P(IR,J-1)=P(IR,J)-T*P(I,J)
  150 P(IR,M)=0.0
      IF (L .EQ. N) GO TO 160
      L=L+1
  160 CONTINUE
C
C     BACK SUBSTITUTION
C
      IF (ABS(P(N,1)) .GE. EPSI) GO TO 400
      WRITE (6,920)N,F(N),P(N,1)
  400 CONTINUE
      F(N)=F(N)/P(N,1)
      JM=2
      DO 180 ICE=1,N1
      IR=N-ICE
      DO 170 J=2,JM
      IRM1=IR-1+J
      F(IR)=F(IR)-P(IR,J)*F(IRM1)
  170 CONTINUE
```

```
      IF (JM .EQ. M) GO TO 180
      JM=JM+1
 180  CONTINUE
C
      RETURN
 920  FORMAT(10X,"N=",I4,5X,"F(N)=",E10.3,5X,"P(N,1)=",E10.3)
       END
      SUBROUTINE BEDHIS(M,T)
C
C     THIS SUBROUTINE KEEPS A RECORD OF BED HISTORY, INCLUDING BED
C     SURFACE ELEVATION, RATIO OF BED SEDIMENT WEIGHT FRACTIONS,
C     AND ASSOCIATED CHEMICAL'S CONCENTRATIONS IN THE BED
C
C     ILAYR(J)....NO. OF LAYERS COMPLETELY SCOURED BY EACH RESPECTIVE
C                 SEDIMENT. ILAYR(J)=-1 FOR DEPOSITION
C     ARAD........AMOUNT OF CHEMICALS LEFT IN THE TOP BED LAYER
C     ALEFT.......AMOUNT OF SEDIMENT LEFT IN THE TOP BED LAYER
C     XDTOP.......TOTAL WEIGHT OF THE SEDIMENT IN THE TOP BED LAYER
C     DRAD........AMOUNT OF CHEMICALS DEPOSITED PER TIME STEP
C     XND.........TOTAL WEIGHT OF THE SEDIMENT IN A BED LAYER
C
      COMMON /BLK1/CT(240),C(9,240)
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
      COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)
      COMMON /BLK12/ACOF(6),U(6),V(6),D(2),AKJ(9,100),ALMBDA,RHOMAT,
     1              AKP(3)
      COMMON /BLK14/GBA(100,10),GBB(100,10),GBC(100,10),GBD(100,10),
     1              GBE(100,10),GBF(100,10),GBG(100,10),POR
      COMMON /BLK15/ILAYR(100,3),XYS0(100),BDIV(100),NBED(100),BED(100),
     1              RHOSED(3),XNT(100,3),RSAV1(100),RSAV2(100),RSAV3(100)
      COMMON /BLK17/GLATE(240),GSARA(7,100),GPNT(7,240),CRATE(3,100),
     1  CD(4,240)
      LEVEL    2,GLATE,GSARA,GPNT,CRATE,CD
      DIMENSION ARAD(3),ALEFT(3),B2(3),B3(3),GBJ(3),CS(3)
      DATA EPSI/1.0E-10/
C
C
      IN=ILAYR(M,1)
      INN=MIN0(IN+1,NBED(M))
      IP=ILAYR(M,2)
      IG=ILAYR(M,3)
C
      GBJ(1) = GBD(M,NBED(M))
      GBJ(2) = GBE(M,NBED(M))
      GBJ(3) = GBF(M,NBED(M))
      CS(1) = RSAV1(M)
      CS(2) = RSAV2(M)
      CS(3) = RSAV3(M)
C
      DO 100 II=1,3
      ARAD(II) = 0.0
      ALEFT(II)=0.0
100   CONTINUE
C
      TOTAL = GBA(M,NBED(M))/RHOSED(1)+GBB(M,NBED(M))/RHOSED(2)+
     $ GBC(M,NBED(M))/RHOSED(3)
      XND=(1.-POR)/TOTAL
      XNTTMP=(XNT(M,1)+XNT(M,2)+XNT(M,3))/XND
      IF(SR(1,M)+SR(2,M)+SR(3,M) .GT. 0.0) GO TO 110
      IF(ABS(XNTTMP-BDIV(M)) .GE. EPSI)     GO TO 110
      XNT(M,1)=0.
      XNT(M,2)=0.
```

```
      XNT(M,3)=0.
      NBED(M)=NBED(M)+1
      DO 105 I=1,3
      ALEFT(I)=SD(I,M)*T
      ARAD(I)=-(CS(I)+CRATE(I,M))*T
  105 CONTINUE
      GO TO 270
110   IF (ILAYR(M,1).LT.0) IN=0
      ALEFT(1)=XNT(M,1)+SD(1,M)*T
      ARAD(1)=XNT(M,1)*GBD(M,NBED(M)-IN)-(CS(1)+CRATE(1,M))*T
      IF (ILAYR(M,2).LT.0) IP=0
      ALEFT(2)=XNT(M,2)+SD(2,M)*T
      ARAD(2)=XNT(M,2)*GBE(M,NBED(M)-IP)-(CS(2)+CRATE(2,M))*T
      IF(ILAYR(M,3) .LT. 0) IQ=0
      ALEFT(3)=XNT(M,3)+SD(3,M)*T
      ARAD(3)=XNT(M,3)*GBF(M,NBED(M)-IQ)-(CS(3)+CRATE(3,M))*T
      IF(ILAYR(M,1) .LT. 1 .OR. ILAYR(M,2) .GE. ILAYR(M,1)) GO TO 150
      IP2=ILAYR(M,2)+2
      DO 140 IT=IP2,INN
      IF (ILAYR(M,2).LT.0.AND.IT.EQ.1) GO TO 140
      IU=NBED(M)-IT+1
      XND=(GBA(M,IU)/RHOSED(1)+GBB(M,IU)/RHOSED(2)+GBC(M,IU)/RHOSED(3))
      XND=(1.0-POR)/XND
      XNT(M,2)=XND*GBB(M,IU)*BDIV(M)
      ALEFT(2)=ALEFT(2)+XNT(M,2)
      ARAD(2)=ARAD(2)+XNT(M,2)*GBE(M,IU)
140   CONTINUE
  150 CONTINUE
      IF(ILAYR(M,1) .LT. 1 .OR. ILAYR(M,3) .GE. ILAYR(M,1)) GO TO 270
      IQ2=ILAYR(M,3)+2
      DO 220 IR=IQ2,INN
      IF (ILAYR(M,3).LT.0.AND.IR.EQ.1) GO TO 220
      IS=NBED(M)-IR+1
      XND=(GBA(M,IS)/RHOSED(1)+GBB(M,IS)/RHOSED(2)+GBC(M,IS)/RHOSED(3))
      XND=(1.0-POR)/XND
      XNT(M,3)=XND*GBC(M,IS)*BDIV(M)
      ALEFT(3)=ALEFT(3)+XNT(M,3)
      ARAD(3)=ARAD(3)+XNT(M,3)*GBF(M,IS)
220   CONTINUE
C
C     XM....THICKNESS OF BED TOP LAYER WHICH  WILL BE SET TO XYSO(M)
C
 270  CONTINUE
      B1=ALEFT(1)+ALEFT(2)+ALEFT(3)
      IF (B1.EQ.0.0.AND.ILAYR(M,1).LT.0) RETURN
      XM = ALEFT(1)/RHOSED(1)+ALEFT(2)/RHOSED(2)+ALEFT(3)/RHOSED(3)
      XM = XM/(1.0-POR)
      IW=0
      REMAIN = XM
280   CONTINUE
      IF (REMAIN.LE.BDIV(M)) GO TO 300
      IW=IW+1
      REMAIN = REMAIN - BDIV(M)
      GO TO 280
300   CONTINUE
      IW=IW+1
      NBED(M) = NBED(M) - INN
      IF (ILAYR(M,1) .LT. 0) NBED(M) = NBED(M) - 1
      NBED1 = NBED(M) + 1
      NBED2 = NBED(M) + IW
      DO 360 IY=NBED1,NBED2
      DO 340 IX=1,3
```

```
      B2(IX)=0.0                                                        BEDHIS   118
      IF (B1.NE.0.0) B2(IX)=ALEFT(IX)/B1                                BEDHIS   119
      B3(IX)=0.0                                                        BEDHIS   120
      IF (ALEFT(IX).GT.0.0.AND.ARAD(IX).GT.0.0) B3(IX)=                 BEDHIS   121
     1 ARAD(IX)/ALEFT(IX)                                               BEDHIS   122
340   CONTINUE                                                          BEDHIS   123
      TONWET=B2(1)*B3(1) + B2(2)*B3(2) + B2(3)*B3(3)                    BEDHIS   124
      GBA(M,IY)=B2(1)                                                   BEDHIS   125
      GBB(M,IY)=B2(2)                                                   BEDHIS   126
      GBC(M,IY)=B2(3)                                                   BEDHIS   127
      GBD(M,IY)=B3(1)* EXP(-ALMBDA*T)                                   BEDHIS   128
      GBE(M,IY)=B3(2)* EXP(-ALMBDA*T)                                   BEDHIS   129
      GBF(M,IY)=B3(3)* EXP(-ALMBDA*T)                                   BEDHIS   130
      GBG(M,IY)=TONWET* EXP(-ALMBDA*T)                                  BEDHIS   131
360   CONTINUE                                                          BEDHIS   132
C                                                                       BEDHIS   133
      NBED(M)=NBED(M)+IM                                                BEDHIS   134
      XYSO(M)=REMAIN                                                    BEDHIS   135
      BED(M)=(NBED(M)-1)*BDIV(M)+XYSO(M)                                BEDHIS   136
      IF (NBED(M).LE.50) GO TO 400                                      BEDHIS   137
      WRITE(6,200)  M,NBED(M)                                           BEDHIS   138
      STOP                                                              BEDHIS   139
400   CONTINUE                                                          BEDHIS   140
      RETURN                                                            BEDHIS   141
200   FORMAT(2X,"DEPOSITION EXCEEDS PERMISSIBLE BED DEPTH IN BEDHIS",/, BEDHIS   142
     15X,"M,NBED=",2I5)                                                 BEDHIS   143
      END                                                               BEDHIS   144
      SUBROUTINE BNDRYS (NROW,NCOL,ID)                                  BNDRYS     2
C                                                                       BNDRYS     3
C     THIS ROUTINE SETS THE BOUNDARY CONDITIONS IN THE GLOBAL MATRICES, BNDRYS     4
C     ELIMINATES THE EQUATIONS FOR BOUNDARY NODES AND THEN REFORMS THE  BNDRYS     5
C     REDUCED SYSTEM MATRICES.                                          BNDRYS     6
C                                                                       BNDRYS     7
      COMMON /BLK1/CT(240),C(9,240)                                     BNDRYS     8
      COMMON /BLK2/P(240,86),F(240)                                     BNDRYS     9
      LEVEL  2,P,F                                                      FETFIX4   11
      COMMON /BLK3/S(240,86)                                            BNDRYS    10
      LEVEL  2,S                                                        FETFIX4   12
      COMMON /BLK4/ R(240), RPAST(240,7), NODBET, BETA1, AREA1          BNDRYS    11
      LEVEL  2,R,RPAST,NODBET,BETA1,AREA1                               FETFIX4   13
      COMMON /BLK8/KBC,LBC,MBC(7,120),NBC(7,120),DBC(7,120),BC(7,120),  BNDRYS    12
     1 KODE(10)                                                         BNDRYS    13
C                                                                       BNDRYS    14
C     SET THE SPECIFIED BOUNDARY CONDITONS IN THE MATRICES              BNDRYS    15
C                                                                       BNDRYS    16
      NBND=2*NCOL-1                                                     BNDRYS    17
      IF (LBC.LE.0) GO TO 190                                           BNDRYS    18
      DO 110 I=1,LBC                                                    BNDRYS    19
      NN=NBC(ID,I)                                                      BNDRYS    20
      R(NN)=0.0                                                         BNDRYS    21
      DO 110 NC=1,NBND                                                  BNDRYS    22
      P(NN,NC)=0.0                                                      BNDRYS    23
      S(NN,NC)=0.0                                                      BNDRYS    24
      NR=NN-NC+NCOL                                                     BNDRYS    25
      IF (NR.LE.0.OR.NR.GT.NROW) GO TO 110                              BNDRYS    26
      R(NR)=R(NR)-S(NR,NC)*BC(ID,I)                                     BNDRYS    27
      P(NR,NC)=0.0                                                      BNDRYS    28
      S(NR,NC)=0.0                                                      BNDRYS    29
110   CONTINUE                                                          BNDRYS    30
C                                                                       BNDRYS    31
C     ELIMINATE THE EQUATIONS FOR THE BOUNDARY CONDITION NODES AND      BNDRYS    32
C     REFORM THE GLOBAL MATRICES.                                       BNDRYS    33
```

```
C
      DO 180 L=1,LBC
      NN=NBC(ID,L)-L+1
      NROW=NROW-1
      IF (NN.GT.NROW) GO TO 180
      DO 120 NR=NN,NROW
      C(ID,NR)=C(ID,NR+1)
      R(NR)=R(NR+1)
      DO 120 NC=1,NCOL
      MC=NCOL+NC-1
      P(NR,MC)=P(NR+1,MC)
      S(NR,MC)=S(NR+1,MC)
      IF (NR.LT.NROW) GO TO 120
      P(NR+1,MC)=0.0
      S(NR+1,MC)=0.0
120   CONTINUE
C
      DO 140 NC=2,NCOL
      MC=NCOL-NC+1
      NL=NN+NC-1
      IF (NL.GT.NROW) GO TO 140
      DO 130 NR=NL,NROW
      P(NR,MC)=P(NR+1,MC)
      S(NR,MC)=S(NR+1,MC)
      IF (NR.LT.NROW) GO TO 130
      P(NR+1,MC)=0.0
      S(NR+1,MC)=0.0
130   CONTINUE
140   CONTINUE
C
      IF (NN.EQ.1) GO TO 180
C
      NCL=NCOL-1
      DO 160 LC=2,NCL
      NR=NN-LC+1
      IF (NR.LE.0) GO TO 160
      DO 150 NC=LC,NCL
      MC=NCOL+NC-1
      P(NR,MC)=P(NR,MC+1)
      S(NR,MC)=S(NR,MC+1)
      IF (NC.LT.NCL) GO TO 150
      P(NR,NBND)=0.0
      S(NR,NBND)=0.0
C     P(NR,MC+1)=0.0
C     S(NR,MC+1)=0.0
150   CONTINUE
160   CONTINUE
C
      NRL=NN+NCOL-3
      DO 170 NR=NN,NRL
      NCL=NN-NR+NCOL-2
      DO 170 NC=1,NCL
      P(NR,NC+1)=P(NR+1,NC)
      S(NR,NC+1)=S(NR+1,NC)
      IF (NC.NE.1) GO TO 170
      P(NR+1,1)=0.0
      S(NR+1,1)=0.0
170   CONTINUE
180   CONTINUE
C
C     PERFORM A LEFT SHIFT OF THE (P)AND (S) MATRIXFOR COMB INNER
C     PRODUCT
```

```
C
190   L1=NCOL-1
      DO 210 IR=1,L1
      LR=NCOL-IR
      DO 210 I=1,LR
      DO 200 J=2,NBND
      P(IR,J-1)=P(IR,J)
      S(IR,J-1)=S(IR,J)
200   CONTINUE
      NP1=NROW+1-IR
      MP1=NBND+1-I
      P(IR,NBND)=0.0
      S(IR,NBND)=0.0
      P(NP1,MP1)=0.0
      S(NP1,MP1)=0.0
210   CONTINUE
C
      RETURN
      END
      SUBROUTINE CLAY (M,ID,T)
C
C     THIS SUBROUTINE COMPUTES THE AMOUNT OF RESUSPENSION, SR,OR
C     DEPOSITION,SD,OF CLAY
C     THIS ROUTINE DECIDES NO.OF BED LAYERS TO BE SCOURED
C     CRSTRS(ID,M).CRITICAL SHEAR STRESS FOR SCOURING,ID=1 FOR CLAY
C     CDSTRS(ID,M).CRITICAL SHEAR STRESS FOR DEPOSITION
C     WS(ID,M).....FALL VELOSITY OF SEDIMENT
C     ERODA(ID,M)..ERODABILITY COEFFICIENT
C     RSAV1........CHEMICAL AMOUNT SCOURED FROM BED DURING ONE TIME STEP
C     RS...........SEDIMENT AMOUNT SCOURED FROM BED DURING ONE TIME STEP
C
      COMMON /BLK1/CT(240),C(9,240)
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
      COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)
      COMMON /BLK13/WS(3,100),CRSTRS(3,100),CDSTRS(3,100),ERODA(3,100)
      COMMON /BLK14/GBA(100,10),GBB(100,10),GBC(100,10),GBD(100,10),
     1              GBE(100,10),GBF(100,10),GBG(100,10),POR
      COMMON /BLK15/ILAYR(100,3),XYS0(100),BDIV(100),NBED(100),BED(100),
     1              RHOSED(3),XNT(100,3),RSAV1(100),RSAV2(100),RSAV3(100)
      DIMENSION S1(3)
      DATA EPSI / 1.0E-10 /
      SD(ID,M)=0.0
      SR(ID,M)=0.0
      RSAV1(M)=0.
      RS=0.0
      ILAYR(M,ID)=0
      TOTAL = GBA(M,NBED(M))/RHOSED(1)+GBB(M,NBED(M))/RHOSED(2)+
     $ GBC(M,NBED(M))/RHOSED(3)
      XDTOP=(1.-POR)/TOTAL
      XNT(M,ID)=XYS0(M)*GBA(M,NBED(M))*XDTOP
      DO 999 II=1,3
      S1(II)=0.0
      IF (STRESS(II).LE.CRSTRS(ID,M).AND.STRESS(II).GE.CDSTRS(ID,M)) GO
     1TO 999
      IF (STRESS(II).GT.CRSTRS(ID,M)) GO TO 100
C     DEPOSITION
      S1(II)=WS(ID,M)*C(ID,NOD(M,II))*(1.0-(STRESS(II)/CDSTRS(ID,M))
     $ )*(-1)
      GO TO 999
C
C
```

```
C      RESUSPENSION
C
100   S1(II)=ERODA(ID,M)*(STRESS(II)/CRSTRS(ID,M)-1.0)
999   CONTINUE
      S2=(S1(1)+S1(2)+S1(3))/3.0
      IF (S2) 300,305,302
300   CONTINUE
C
C     DEPOSITION
C
      SD(ID,M)=(-1)*S2
      ILAYR(M,ID)=-1
      N1 = NOD(M,1)
      N2 = NOD(M,2)
      N3 = NOD(M,3)
      N4 = NOD(M,4)
      N5 = NOD(M,5)
      N6 = NOD(M,6)
      H1 = H(N1)
      H2 = H(N2)
      H3 = H(N3)
      HAVG = (H1+H2+H3)/3.
      CAVG1 = (C(ID,N1)*H1+C(ID,N2)*H2+C(ID,N3)*H3+C(ID,N4)*(H1+H2)/2.
     1  +C(ID,N5)*(H2+H3)/2.+C(ID,N6)*(H1+H3)/2.)/(6.*HAVG)
      IF (CAVG1.LE.EPSI) GO TO 305
      CAVG5 = (C(ID+4,N1)*H1+C(ID+4,N2)*H2+C(ID+4,N3)*H3+C(ID+4,N4)*
     1  (H1+H2)/2. C(ID+4,N5)*(H2+H3)/2.+C(ID+4,N6)*(H1+H3)/2.)/
     2  (6.*HAVG)
      RSAV1(M) = -SD(ID,M)*CAVG5/CAVG1
      GO TO 305
302   CONTINUE
C
C**       RESUSPENSION
C
      SR(ID,M)=S2
C     TO COMPUTE A NUMBER OF LAYERS SCOURED IN ORDER TO RESUSPEND THE
C     APPROPRIATE AMOUNT OF COHESIVE SEDIMENT
C     ASSUME THAT CLAY IS MOST DIFFICULT TO SCOUR
C     XNT(M)....WEIGHT OF THE SEDIMENT  M IN THE TOP BED LAYER IN KG.
C     NBED(M)...INITIAL NUMBER OF BED LAYERS IN ELEMENT M
C     BED(M)....INITIAL BED THICKNESS IN ELEMENT M
C     XYS0(M)...THICKNESS OF THE TOP BED LAYER IN ELEMENT M
C     GBA(M,J)..WEIGHT FRACTION OF CLAY OR SAND2 OF BED LAYER J IN
C      ELEMENT M
C     GBB(M,J)  WEIGHT FRACTION OF SILT OF BED LAYER J IN ELEMENT M
C     GBC(M,J)  WEIGHT FRACTION OF SAND OF BED LAYER J IN ELEMENT M
C     GBD(M,J)  CHEMICAL CONCENTRATION IN CLAY OF BED LAYER J IN ELEM. M
C               PER UNIT WEIGHT OF SEDIMENT
C     GBE(M,J)  CHEMICAL CONCENTRATION IN SILT OF BED LAYER J IN ELEM. M
C               PER UNIT WEIGHT OF SEDIMENT
C     GBF(M,J)  CHEMICAL CONCENTRATION IN SAND OF BED LAYER [illegible] ELEM. M
C               PER UNIT WEIGHT OF SEDIMENT
C     GBG(M,J)  TOTAL CHEMICAL CONCENTRATION IN BED LAYER [illegible] ELEMENT M
C               PER UNIT WEIGHT OF SEDIMENT
C     GBG(M,J)=GBA(M,J)*GBD(M,J)+GBB(M,J)*GBE(M,J)+[illegible]M,J)
C
      NB = NBED(M)
      SR(ID,M)=SR(ID,M)*T
      IF (SR(ID,M).GT.XNT(M,ID)) GO TO 200
      RSAV1(M)=SR(ID,M)*GBD(M,NBED(M))
      XNT(M,ID)=XNT(M,ID)-SR(ID,M)
      GO TO 290
```

```
C       ILAYR(J)..NO.OF TOTAL BED LAYERS SCOURED DURING TIME DURATION,T    CLAY     107
C       BDIV(M)...THICKNESS OF BED LAYER IN ELEMENT M                      CLAY     108
200     ILAYR(M,ID)=ILAYR(M,ID)+1                                          CLAY     109
        IF (ILAYR(M,ID).EQ.NBED(M)) GO TO 280                              CLAY     110
        RSAV1(M)=RSAV1(M)+XNT(M,ID)*GBD(M,NB)                              CLAY     111
        SR(ID,M)=SR(ID,M)-XNT(M,ID)                                        CLAY     112
        RS=RS+XNT(M,ID)                                                    CLAY     113
        NB=NBED(M)-ILAYR(M,ID)                                             CLAY     114
        TOTAL=GBA(M,NB)/RHOSED(1)+GBB(M,NB)/RHOSED(2)+GBC(M,NB)/RHOSED(3)  CLAY     115
        XND=(1.-POR)/TOTAL                                                 CLAY     116
        XNT(M,ID)=BDIV(M)*GBA(M,NB)*XND                                    CLAY     117
        IF (ILAYR(M,ID).LT.(NBED(M)-1)) GO TO 270                          CLAY     118
        ILAYR(M,ID)=ILAYR(M,ID)+1                                          CLAY     119
        GO TO 280                                                          CLAY     120
270     IF (SR(ID,M).GE.XNT(M,ID)) GO TO 200                               CLAY     121
280     RSAV1(M)=RSAV1(M)+AMIN1(SR(ID,M),XNT(M,ID))*GBD(M,NB)              CLAY     122
        RS=RS+AMIN1(SR(ID,M),XNT(M,ID))                                    CLAY     123
        XNT(M,ID)=XNT(M,ID)-AMIN1(SR(ID,M),XNT(M,ID))                      CLAY     124
        SR(ID,M)=RS                                                        CLAY     125
290     SR(ID,M)=SR(ID,M)/T                                                CLAY     126
        RSAV1(M)=RSAV1(M)/T                                                CLAY     127
  305 CONTINUE                                                             CLAY     128
        RETURN                                                             CLAY     129
        END                                                                CLAY     130
        SUBROUTINE COMB(NROW,NCOL,ID)                                      COMB       2
C                                                                          COMB       3
C       THIS SUBROUTINE MULTIPLIES THE UNSYMMETRIC BAND MATRIX (S) TO THE  COMB       4
C       LOAD VECTOR (C) AND STORES THE RESULT IN (F).                      COMB       5
        DOUBLE PRECISION DVAR                                              COMB       6
        COMMON /BLK1/CT(240),C(9,240)                                      COMB       7
        COMMON /BLK2/P(240,86),F(240)                                      COMB       8
        LEVEL  2,P,F                                                       FETFIX4   14
        COMMON /BLK3/S(240,86)                                             COMB       9
        LEVEL  2,S                                                         FETFIX4   15
C                                                                          COMB      10
        NBND=2*NCOL-1                                                      COMB      11
        DO 100 I=1,NCOL                                                    COMB      12
        F(I)=0.0                                                           COMB      13
        DO 100 K=1,NBND                                                    COMB      14
        DVAR=F(I)+S(I,K)*C(ID,K)                                           COMB      15
        F(I)=DVAR                                                          COMB      16
 100    CONTINUE                                                           COMB      17
C                                                                          COMB      18
        LL=0                                                               COMB      19
        NI=NCOL+1                                                          COMB      20
        IF (NI .GT. NROW) RETURN                                           COMB      21
        DO 110 I=NI,NROW                                                   COMB      22
        F(I)=0.0                                                           COMB      23
        LL=LL+1                                                            COMB      24
        DO 110 K=1,NBND                                                    COMB      25
        L=LL+K                                                             COMB      26
        IF (L .GT. NROW) GO TO 110                                         COMB      27
        DVAR=F(I)+S(I,K)*C(ID,L)                                           COMB      28
        F(I)=DVAR                                                          COMB      29
 110    CONTINUE                                                           COMB      30
C        IF(ID.LT.5) GO TO 995                                             COMB      31
C        WRITE(6,996)                                                      COMB      32
C        WRITE(6,997) (F(I),I=1,NROW)                                      COMB      33
C        WRITE(6,998)                                                      COMB      34
C        WRITE(6,999) (C(ID,I),I=1,NROW)                                   COMB      35
C996    FORMAT(//,10X,"F MATRIX IN COMB",/)                                COMB      36
C997    FORMAT(12(1PE10.3))                                                COMB      37
```

```
C998   FORMAT(//,10X,"C IN COMB",/)
C999   FORMAT(12(1PE10.3))
C995   CONTINUE
C
      RETURN
      END
      SUBROUTINE DISOLV (M,AREA,ID,IDN,T)
C
C     THIS ROUTINE CALCULATES COEFFICIENTS OF DECAY AND
C     SOURCE TERMS IN THE DISSOLVED CHEMICAL TRANSPORT CONVECTION-
C     DIFFUSION EQUATION
C
      COMMON /BLK1/CT(240),C(9,240)
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
      COMMON /BLK9/DX(100),DY(100),EX(100),EY(100),ALFA(100),BETA(100)
     9          ,HS(100)
      COMMON /BLK10/PEL(6,6),SEL(6,6),REL(6)
      COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)
      COMMON /BLK12/ACOF(6),U(6),V(6),D(2),AKJ(9,100),ALMBDA,RHOMAT,
     1              AKP(3)
      COMMON /BLK14/GBA(100,10),GBB(100,10),GBC(100,10),GBD(100,10),
     1              GBE(100,10),GBF(100,10),GBG(100,10),POR
      COMMON /BLK15/ILAYR(100,3),XYS0(100),BDIV(100),NBED(100),BED(100),
     1             RHOSED(3),XNT(100,3),RSAV1(100),RSAV2(100),RSAV3(100)
      COMMON /BLK17/QLATE(240),QSARA(7,100),QPNT(7,240),CRATE(3,100),
     1  CD(4,240)
      LEVEL   2,QLATE,QSARA,QPNT,CRATE,CD
      DIMENSION GBJ(3)
      DIMENSION GBI(3)
C
      DATA EPSI/1.0E-12/
      A1=ACOF(1)
      A2=ACOF(2)
      A3=ACOF(3)
      B1=ACOF(4)
      B2=ACOF(5)
      B3=ACOF(6)
C
      N1=NOD(M,1)
      N2=NOD(M,2)
      N3=NOD(M,3)
      N4=NOD(M,4)
      N5=NOD(M,5)
      N6=NOD(M,6)
C
C
      H1        = H(N1)
      H2        = H(N2)
      H3        = H(N3)
C
C***     DECAY TERM      ***
C
      ALFA(M)  = ALMBDA
C**   ALFA(M)  = ALMBDA+QLATE(M) IF QLATE(M) IS CONSTANT IN A ELEMENT
C
C***     SOURCE OR SINK TERM      ***
C
      DO 100 I = 1,6
      REL(I)   = 0.0
100   CONTINUE
C
```

```
C       ADSORPTION/DESORPTION WITH NON-MOVING BED SEDIMENT                    DISOLV    56
C                                                                             DISOLV    57
        GBI(1)     = GBA(M,NBED(M))                                           FETFIX8    2
        GBI(2)     = GBB(M,NBED(M))                                           FETFIX8    3
        GBI(3)     = GBC(M,NBED(M))                                           FETFIX8    4
        GBJ(1)     = GBD(M,NBED(M))                                           DISOLV    58
        GBJ(2)     = GBE(M,NBED(M))                                           DISOLV    59
        GBJ(3)     = GBF(M,NBED(M))                                           DISOLV    60
        A          = 0.0                                                      DISOLV    61
        B          = 0.0                                                      DISOLV    62
        DO 105 I = 1,3                                                        DISOLV    63
        AA         = RHOSED(I)*(1.0-POR)*BD50(M)*AKJ(I+6,M)                   FETFIX6   26
        AA         = AA*GBI(I)                                                FETFIX8    5
        A          = -AA*AKJ(I,M)                                             DISOLV    65
        B          = AA*GBJ(I)                                                DISOLV    66
        CRATE(I,M) = 0.0                                                      DISOLV    67
        IF (SR(ID,M).GT.0) GO TO 105                                          DISOLV    68
        IF (AKJ(I,M).EQ.0.OR.AKJ(I+6,M).EQ.0) GO TO 105                       DISOLV    69
        A          = A*AREA/180.                                              DISOLV    70
        B          = B*AREA/3.                                                DISOLV    71
        REL(1)     = RE (1) +A*(6.*C(ID,N1)-C(ID,N2)-C(ID,N3)-4.*C(ID,N5))    DISOLV    72
        REL(2)     = REL(2) -A*(C(ID,N1)-6.*C(ID,N2)+C(ID,N3)+4.*C(ID,N6))    DISOLV    73
        REL(3)     = REL(3) -A*(C(ID,N1)+C(ID,N2)-6.*C(ID,N3)+4.*C(ID,N4))    DISOLV    74
        REL(4)     = REL(4) -A*(4.*C(ID,N3)-32.*C(ID,N4)-16.*C(ID,N5)         DISOLV    75
       1                     -16.*C(ID,N6))+B                                 DISOLV    76
        REL(5)     = REL(5) -A*(4.*C(ID,N1)-16.*C(ID,N4)-32.*C(ID,N5)         DISOLV    77
       1                     -16.*C(ID,N6))+B                                 DISOLV    78
        REL(6)     = REL(6) -A*(4.*C(ID,N2)-16.*C(ID,N4)-16.*C(ID,N5)         DISOLV    79
       1                     -32.*C(ID,N6))+B                                 DISOLV    80
        CRATE(I,M) = (REL(1)+REL(2)+REL(3)+REL(4)+REL(5)+REL(6))              DISOLV    81
C                                                                             DISOLV    82
C                                                                             DISOLV    83
C       IF (M.NE.3) GO TO 105                                                 DISOLV    84
C200    FORMAT(//,2X,"A = ",E12.3,4X,"REL(I) = ",6E12.3,/)                    DISOLV    85
C201    FORMAT(//,2X,"CRATE(I) = ",E12.3,4X,"C(ID,NI) =",6E12.3,/)            DISOLV    86
C       WRITE(6,200) A,(REL(J),J=1,6)                                         DISOLV    87
C       WRITE(6,201)  CRATE(I,M),C(ID,N1),C(ID,N2),C(ID,N3),C(ID,N4),         DISOLV    88
C      1  C(ID,N5),C(ID,N6)                                                   DISOLV    89
  105   CONTINUE                                                              DISOLV    90
        CRATE(3,M) = CRATE(3,M)-CRATE(2,M)                                    FETFIX6   27
        CRATE(2,M) = CRATE(2,M)-CRATE(1,M)                                    DISOLV    91
        CRATE(1,M) = CRATE(1,M)/AREA                                          DISOLV    93
        CRATE(2,M) = CRATE(2,M)/AREA                                          DISOLV    94
        CRATE(3,M) = CRATE(3,M)/AREA                                          DISOLV    95
C                                                                             DISOLV    96
C                                                                             DISOLV    97
C       DISSOLVED CONTAMINANT SOURCE                                          DISOLV    98
C                                                                             DISOLV    99
        A          = AREA*QSARA(ID,M)/60.                                     DISOLV   100
C                                                                             DISOLV   101
        REL(1)     = A*(2.*H1-H2-H3)          +REL(1)                         DISOLV   102
        REL(2)     = -A*(H1-2.*H2+H3)         +REL(2)                         DISOLV   103
        REL(3)     = -A*(H1+H2-2.*H3)         +REL(3)                         DISOLV   104
        REL(4)     = A*(8.*H1+8.*H2+4.*H3)    +REL(4)                         DISOLV   105
        REL(5)     = A*(4.*H1+8.*H2+8.*H3)    +REL(5)                         DISOLV   106
        REL(6)     = A*(8.*H1+4.*H2+8.*H3)    +REL(6)                         DISOLV   107
C                                                                             DISOLV   108
C                                                                             DISOLV   109
C       ADSORPTION/DESORPTION WITH MOVING SEDIMENT                            DISOLV   110
C                                                                             DISOLV   111
        DO 110 I = 1,3                                                        DISOLV   112
        A          = AKJ(I,M)*AKJ(I+3,M)*AREA/180.                            DISOL    113
```

```
      B        = AKJ(I+3,M)*AREA/1260.
C
      REL(1)    = REL(1)-A*(6.*H1*CD(I,N1)*C(ID,N1)-H2*CD(I,N2)*C(ID,N2)
     1    -H3*CD(I,N3)*C(ID,N3)-2.*(H2+H3)*CD(I,N5)*C(ID,N5))
      REL(2)    = REL(2)+A*(H1*CD(I,N1)*C(ID,N1)-6.*H2*CD(I,N2)*C(ID,N2)
     1    +H3*CD(I,N3)*C(ID,N3)+2.*(H1+H3)*CD(I,N6)*C(ID,N6))
      REL(3)    = REL(3)+A*(H1*CD(I,N1)*C(ID,N1)+H2*CD(I,N2)*C(ID,N2)
     1    -6.*H3*CD(I,N3)*C(ID,N3)+2.*(H1+H2)*CD(I,N4)*C(ID,N4))
      REL(4)    = REL(4)+A*(4.*H3*CD(I,N3)*C(ID,N3)-16.*(H1+H2)*CD(I,N4)*
     1    C(ID,N4)-6.*(H2+H3)*CD(I,N5)*C(ID,N5)-8.*(H1+H3)*CD(I,N6)*
     2    C(ID,N6))
      REL(5)    = REL(5)+A*(4.*H1*CD(I,N1)*C(ID,N1)-8.*(H1+H2)*CD(I,N4)*
     1    C(ID,N4)-16.*(H2+H3)*CD(I,N5)*C(ID,N5)-8.*(H1+H3)*CD(I,N6)*
     2    C(ID,N6))
      REL(6)    = REL(6)+A*(4.*H2*CD(I,N2)*C(ID,N2)-8.*(H1+H2)*CD(I,N4)*
     1    C(ID,N4)-8.*(H2+H3)*CD(I,N5)*C(ID,N5)-16.*(H1+H3)*CD(I,N6)*
     2    C(ID,N6))
C
      REL(1)    = REL(1)              +B*((30.*H1+6.*H2+6.*H3)*C(ID+I,N1)
     1-(4.*H1+4.*H2-H3)*C(ID+I,N2)      -(4.*H1-H2+4.*H3)*C(ID+I,N3)
     2+(12.*H1-8.*H2-4.*H3)*C(ID+I,N4)-(4.*H1+12.*H2+12.*H3)*C(ID+I,N5)
     3+(12.*H1-4.*H2-8.*H3)*C(ID+I,N6))
      REL(2)    = REL(2)              +B*((4.*H1+4.*H2-H3)*C(ID+I,N1)*(-1)
     1+(6.*H1+30.*H2+6.*H3)*C(ID+I,N2)+(H1-4.*H2-4.*H3)*C(ID+I,N3)
     2-(8.*H1-12.*H2+4.*H3)*C(ID+I,N4)-(4.*H1-12.*H2+8.*H3)*C(ID+I,N5)
     3-(12.*H1+4.*H2+12.*H3)*C(ID+I,N6))
      REL(3)    = REL(3)              +B*((4.*H1-H2+4.*H3)*C(ID+I,N1)*(-1)
     1+(H1-4.*H2-4.*H3)*C(ID+I,N2)      +(6.*H1+6.*H2+30.*H3)*C(ID+I,N3)
     2-(12.*H1+12.*H2+4.*H3)*C(ID+I,N4)-(4.*H1+8.*H2-12.*H3)*C(ID+I,N5)
     3-(8.*H1+4.*H2-12.*H3)*C(ID+I,N6))
      REL(4)    = REL(4)              +B*((12.*H1-8.*H2-4.*H3)*C(ID+I,N1)
     1-(8.*H1-12.*H2+4.*H3)*C(ID+I,N2)-(12.*H1+12.*H2+4.*H3)*C(ID+I,N3)
     2+(96.*H1+96.*H2+32.*H3)*C(ID+I,N4)+(32.*H1+48.*H2+32.*H3)*
     3C(ID+I,N5)+(48.*H1+32.*H2+32.*H3)*C(ID+I,N6))
      REL(5)    = REL(5)              +B*((4.*H1+12.*H2+12.*H3)*C(ID+I,N1)*(-1)
     1-(4.*H1-12.*H2+8.*H3)*C(ID+I,N2)-(4.*H1+8.*H2-12.*H3)*C(ID+I,N3)
     2+(32.*H1+48.*H2+32.*H3)*C(ID+I,N4)+(32.*H1+96.*H2+96.*H3)*
     3C(ID+I,N5)+(32.*H1+32.*H2+48.*H3)*C(ID+I,N6))
      REL(6)    = REL(6)              +B*((12.*H1-4.*H2-8.*H3)*C(ID+I,N1)
     1-(12.*H1+4.*H2+12.*H3)*C(ID+I,N2)-(8.*H1+4.*H2-12.*H3)*C(ID+I,N3)
     2+(48.*H1+32.*H2+32.*H3)*C(ID+I,N4)+(32.*H1+32.*H2+48.*H3)*
     3C(ID+I,N5)+(96.*H1+32.*H2+96.*H3)*C(ID+I,N6))
110   CONTINUE
C
C     POINT SOURCE QPNT(ID,NDS)
C
      B        = 1.0
C
      REL(1)    = REL(1)              +B*((30.*H1+6.*H2+6.*H3)*QPNT(ID,N1)
     1-(4.*H1+4.*H2-H3)*QPNT(ID,N2)      -(4.*H1-H2+4.*H3)*QPNT(ID,N3)
     2+(12.*H1-8.*H2-4.*H3)*QPNT(ID,N4)-(4.*H1+12.*H2+12.*H3)*
     3QPNT(ID,N5)+(12.*H1-4.*H2-8.*H3)*QPNT(ID,N6))
      REL(2)    = REL(2)              +B*((4.*H1+4.*H2-H3)*QPNT(ID,N1)*(-1)
     1+(6.*H1+30.*H2+6.*H3)*QPNT(ID,N2)+(H1-4.*H2-4.*H3)*QPNT(ID,N3)
     2-(8.*H1-12.*H2+4.*H3)*QPNT(ID,N4)-(4.*H1-12.*H2+8.*H3)*QPNT(ID,N5)
     3-(12.*H1+4.*H2+12.*H3)*QPNT(ID,N6))
      REL(3)    = REL(3)              +B*((4.*H1-H2+4.*H3)*QPNT(ID,N1)*(-1)
     1+(H1-4.*H2-4.*H3)*QPNT(ID,N2)      +(6.*H1+6.*H2+30.*H3)*QPNT(ID,N3)
     2-(12.*H1+12.*H2+4.*H3)*QPNT(ID,N4)-(4.*H1+8.*H2-12.*H3)*
     3QPNT(ID,N5)-(8.*H1+4.*H2-12.*H3)*QPNT(ID,N6))
      REL(4)    = REL(4)              +B*((12.*H1-8.*H2-4.*H3)*QPNT(ID,N1)
     1-(8.*H1-12.*H2+4.*H3)*QPNT(ID,N2)-(12.*H1+12.*H2+4.*H3)*
```

```
      2QPNT(ID,N3)+(96.*H1+96.*H2+32.*H3)*QPNT(ID,N4)+(32.*H1+48.*H2+32.*
      3H3)*QPNT(ID,N5)+(48.*H1+32.*H2+32.*H3)*QPNT(ID,N6))
       REL(5)    = REL(5)    +8*((4.*H1+12.*H2+12.*H3)*QPNT(ID,N1)*(-1)
      1-(4.*H1-12.*H2+8.*H3)*QPNT(ID,N2)-(4.*H1+8.*H2-12.*H3)*QPNT(ID,N3)
      2+(32.*H1+48.*H2+32.*H3)*QPNT(ID,N4)+(32.*H1+96.*H2+96.*H3)*
      3QPNT(ID,N5)+(32.*H1+32.*H2+48.*H3)*QPNT(ID,N6))
       REL(6)    = REL(6)              +8*((12.*H1-4.*H2-8.*H3)*QPNT(ID,N1)
      1-(12.*H1+4.*H2+12.*H3)*QPNT(ID,N2)-(8.*H1+4.*H2-12.*H3)*
      2QPNT(ID,N3)+(48.*H1+32.*H2+32.*H3)*QPNT(ID,N4)+(32.*H1+32.*H2+48.
      3*H3)*QPNT(ID,N5)+(96.*H1+32.*H2+96.*H3)*QPNT(ID,N6))
C
       RETURN
       END
       SUBROUTINE DUBOY(ID,M,II,QS)
C
C**    THIS SUBROUTINE COMPUTES THE SEDIMENT LOAD
C**    CAPACITY BY DUBOY'S FORMULA
C
       COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)
       COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)
C
C
C      DUBOY'S FORMULA MUST USE THE FOLLOWING UNITS
C      D50(J,M)..MEDIAN SAND DIAMETER IN MILLIMETERS  J=1,2,3   M=1,NE
C      QS........TOTAL SAND LOAD PER UNIT WIDTH IN LB/SEC-FT
C      STRESS(J).BED SHEAR STRESS IN LB/SQ.FT
C      TAUC......CRITICAL SHEAR STRESS AT WHICH SEDIMENT
C                MOVEMENT BEGINS, LBF/FT**2.
C
       J=ID
       D50TMP  =D50(J,M)*1000.0
       TAU=0.2627*ALOG10(D50TMP  )*ALOG10(D50TMP  )+0.590*ALOG10(D50TMP
      1)-1.4962
       TAUC=10.0**(TAU)
       PSI=ALOG10(28.8)-0.7365*ALOG10(D50TMP)
       PSID=10.0**(PSI)
       QS=0.
       STRESS(II)=STRESS(II)/4.88243
       IF (STRESS(II).LE.TAUC) GO TO 999
       QS=PSID*STRESS(II)*(STRESS(II)-TAUC)
       QS=QS*128577.0
   999 RETURN
       END
       SUBROUTINE ERFC(X,ANS)
C
C**    THIS SUBROUTINE CALCULATES THE ERROR FUNCTION OF ABS(X) TO WITHIN
C**    3*10**-7.  IT THEN CALCULATES THE COMPLIMENTARY ERROR FUNCTION
C**    OF X AND RETURNS THAT VALUE AS ANS.  ANS IS VALID FOR ALL REAL X.
C
       DIMENSION A(6)
       DATA A(1),A(2),A(3),A(4),A(5),A(6)/.0705230784,.0422820123,
      $.0092705272,.0001520143,.0002765672,.0000430638/
C
C**    NOW CALCULATE ERF(X)
C
       XX=1.
       ACCUM=1.
       DO 10 I=1,6
       XX=ABS(X)*XX
       ACCUM=ACCUM+XX*A(I)
    10 CONTINUE
C
```

```
C**   THIS CHECK WORKS ON THE CDC 7000 SERIES.
C
C     IF(ACCUM .GT. 10.**20) GO TO 20
C
C**   THIS CHECK WORKS ON THE VAX 11/780.
C
      IF(ACCUM .GT. 200.0) GO TO 20
      ERF=1.-1./ACCUM**16
      GO TO 30
   20 ERF=1.
C
C**   NOW CALCULATE ANS = ERFC(X)
C
   30 ANS=1.-ERF
      IF (X .GE. 0.) RETURN
      ANS=1.+ERF
      RETURN
      END
      SUBROUTINE MATADD (M,NCOL,ID)
C
C     THIS SUBROUTINE ASSEMBLES THE GLOBAL SYSTEM MATRICES (P),(S) AND
C     THE LOAD VECTOR R BY ADDING IN EACH ELEMENTAL MATRIX. THE
C     SYSTEM MATRICES ARE STORED IN BAND FORM
C
      COMMON /BLK2/P(240,86),F(240)
      LEVEL  2,P,F
      COMMON /BLK3/S(240,86)
      LEVEL  2,S
      COMMON /BLK4/ R(240), RPAST(240,7), NODBET, BETA1, AREA1
      LEVEL  2,R,RPAST,NODBET,BETA1,AREA1
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
      COMMON /BLK10/PEL(6,6),SEL(6,6),REL(6)
      COMMON /BLK12/ACOF(6),U(6),V(6),D(2),AKJ(9,100),ALMBDA,RHOMAT,
     1                AKP(3)
C
C
      NBND=2*NCOL-1
      DO 110 J=1,6
      NR=NOD(M,J)
      DO 100 K=1,6
      NC=NOD(M,K)-NR+1
      MC=NCOL+NC-1
      P(NR,MC)=P(NR,MC)+PEL(J,K)
      S(NR,MC)=S(NR,MC)+SEL(J,K)
100   CONTINUE
      R(NR)=R(NR)+REL(J)
110   CONTINUE
      IF (ID.EQ.4.AND.M.EQ.1) R(NODBET) = R(NODBET)+BETA1
C     IF (ID.EQ.5.AND.M.EQ.182) R(NODBET)=R(NODBET)+AKJ(1,M)*BETA1
C     IF (ID.EQ.6.AND.M.EQ.182) R(NODBET)=R(NODBET)+AKJ(2,M)*BETA1
C     IF (ID.EQ.7.AND.M.EQ.182) R(NODBET)=R(NODBET)+AKJ(3,M)*BETA1
C
      RETURN
      END
      SUBROUTINE PARTIC (M,AREA,ID,IDN,T)
C
C     THIS SUBROUTINE CALCULATES COEFFICIENTS OF
C     DECAY AND SOURCE TERMS IN THE PARTICULATE NUCLIDE TRANSPORT EQ.
C
      COMMON /BLK1/CT(240),C(9,240)
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
```

```
      COMMON /BLK9/DX(100),DY(100),EX(100),EY(100),ALFA(100),BETA(100)
     9             ,HS(100)
      COMMON /BLK10/PEL(6,6),SEL(6,6),REL(6)
      COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)
      COMMON /BLK12/ACOF(6),U(6),V(6),D(2),AKJ(9,100),ALMBDA,RHOMAT,
     1              AKP(3)
      COMMON /BLK14/GBA(100,10),GBB(100,10),GBC(100,10),GBD(100,10),
     1              GBE(100,10),GBF(100,10),GBG(100,10),POR
      COMMON /BLK15/ILAYR(100,3),XYS0(100),BDIV(100),NBED(100),BED(100),
     1             RHOSED(3),XNT(100,3),RSAV1(100),RSAV2(100),RSAV3(100)
      COMMON /BLK17/QLATE(240),QSARA(7,100),QPNT(7,240),CRATE(3,100),
     1  CD(4,240)
      LEVEL   2,QLATE,QSARA,QPNT,CRATE,CD
      DIMENSION CS(3)
      DATA EPSI/1.0E-12/
C
      A1=ACOF(1)
      A2=ACOF(2)
      A3=ACOF(3)
      B1=ACOF(4)
      B2=ACOF(5)
      B3=ACOF(6)
C
      N1=NOD(M,1)
      N2=NOD(M,2)
      N3=NOD(M,3)
      N4=NOD(M,4)
      N5=NOD(M,5)
      N6=NOD(M,6)
      H1        = H(N1)
      H2        = H(N2)
      H3        = H(N3)
C
C
C***      DECAY TERM      ***
C
      ALFA(M)   = ALMBDA
C**   ALFA(M)   = ALMBDA+QLATE(M) IF QLATE(M)IS CONSTANT IN A ELEMENT
C     QLATE(M) IS A LATERAL INFLOW OF WATER PER UNIT VOLUME (1/SEC)
C
C
C***      SOURCE OR SINK TERM      ***
C
      DO 100 I = 1,6
      REL(I)    = 0.0
100   CONTINUE
C
C     DUE TO SEDIMENT EROSION AND/OR DEPOSITION
C
      CS(1)     = RSAV1(M) * AREA/3.0
      CS(2)     = RSAV2(M) * AREA/3.0
      CS(3)     = RSAV3(M) * AREA/3.0
      REL(1)    = 0.0
      REL(2)    = 0.0
      REL(3)    = 0.0
      REL(4)    = CS(ID-4)
      REL(5)    = REL(4)
      REL(6)    = REL(4)
C
C         IF (M.EQ.77) WRITE(6,999) ID,(REL(I),I=1,6)
C999      FORMAT(2X,"ID,REL= ",I5,6E12.5)
C     SORBED CONTAMINANT GENERATION
```

```
C
      A          = AREA*QSARA(ID,M)/60.
      REL(1)     = REL(1)+A*(2.*H1-H2-H3)
      REL(2)     = REL(2)-A*(H1-2.*H2+H3)
      REL(3)     = REL(3)-A*(H1+H2-2.*H3)
      REL(4)     = REL(4)+A*(8.*H1+8.*H2+4.*H3)
      REL(5)     = REL(5)+A*(4.*H1+8.*H2+8.*H3)
      REL(6)     = REL(6)+A*(8.*H1+4.*H2+8.*H3)
C         IF(M.EQ.77) WRITE(6,999) ID,(REL(I),I=1,6)
C
      I=ID-4
      A          = AKJ(I,M)*AKJ(I+3,M)*AREA/180.
      B          = AKJ(I+3,M)*AREA/1260.
C
      REL(1)     = REL(1)+A*(6.*H1*CD(I,N1)*CD(4,N1)-H2*CD(I,N2)*CD(4,N2)
     1    -H3*CD(I,N3)*CD(4,N3)-2.*(H2+H3)*CD(I,N5)*CD(4,N5))
      REL(2)     = REL(2)-A*(H1*CD(I,N1)*CD(4,N1)-6.*H2*CD(I,N2)*CD(4,N2)
     1    +H3*CD(I,N3)*CD(4,N3)+2.*(H1+H3)*CD(I,N6)*CD(4,N6))
      REL(3)     = REL(3)-A*(H1*CD(I,N1)*CD(4,N1)+H2*CD(I,N2)*CD(4,N2)
     1    -6.*H3*CD(I,N3)*CD(4,N3)+2.*(H1+H2)*CD(I,N4)*CD(4,N4))
      REL(4)     = REL(4)-A*(4.*H3*CD(I,N3)*CD(4,N3)-16.*(H1+H2)*CD(I,N4)*
     1    CD(4,N4)-8.*(H2+H3)*CD(I,N5)*CD(4,N5)-8.*(H1+H3)*CD(I,N6)*
     2    CD(4,N6))
      REL(5)     = REL(5)-A*(4.*H1*CD(I,N1)*CD(4,N1)-8.*(H1+H2)*CD(I,N4)*
     1    CD(4,N4)-16.*(H2+H3)*CD(I,N5)*CD(4,N5)-8.*(H1+H3)*CD(I,N6)*
     2    CD(4,N6))
      REL(6)     = REL(6)-A*(4.*H2*CD(I,N2)*CD(4,N2)-8.*(H1+H2)*CD(I,N4)*
     1    CD(4,N4)-8.*(H2+H3)*CD(I,N5)*CD(4,N5)-16.*(H1+H3)*CD(I,N6)*
     2    CD(4,N6))
C
C         IF(M.EQ.77) WRITE(6,999) ID,(REL(J),J=1,6)
C       IF(M.EQ.77) WRITE(6,998) N2,I,H1,H2,H3,CD(I,N2),CD(4,N2),C(ID,N2)
C998       FORMAT(2X,"N2,I,H,CD(I,4,ID,N2)",2I5,6E12.5)
      REL(1)     = REL(1)                -B*((30.*H1+6.*H2+6.*H3)*C(ID,N1)
     1-(4.*H1+4.*H2-H3)*C(ID,N2)     -(4.*H1-H2+4.*H3)*C(ID,N3)
     2+(12.*H1-8.*H2-4.*H3)*C(ID,N4)-(4.*H1+12.*H2+12.*H3)*C(ID,N5)
     3+(12.*H1-4.*H2-8.*H3)*C(ID,N6))
      REL(2)     = REL(2)                -B*((4.*H1+4.*H2-H3)*C(ID,N1)*(-1)
     1+(6.*H1+30.*H2+6.*H3)*C(ID,N2)+(H1-4.*H2-4.*H3)*C(ID,N3)
     2-(8.*H1-12.*H2+4.*H3)*C(ID,N4)-(4.*H1-12.*H2+8.*H3)*C(ID,N5)
     3-(12.*H1+4.*H2+12.*H3)*C(ID,N6))
      REL(3)     = REL(3)                -B*((4.*H1-H2+4.*H3)*C(ID,N1)*(-1)
     1+(H1-4.*H2-4.*H3)*C(ID,N2)     +(6.*H1+6.*H2+30.*H3)*C(ID,N3)
     2-(12.*H1+12.*H2+4.*H3)*C(ID,N4)-(4.*H1+8.*H2-12.*H3)*C(ID,N5)
     3-(8.*H1+4.*H2-12.*H3)*C(ID,N6))
      REL(4)     = REL(4)                -B*((12.*H1-8.*H2-4.*H3)*C(ID,N1)
     1-(8.*H1-12.*H2+4.*H3)*C(ID,N2)-(12.*H1+12.*H2+4.*H3)*C(ID,N3)
     2+(96.*H1+96.*H2+32.*H3)*C(ID,N4)+(32.*H1+48.*H2+32.*H3)*
     3C(ID,N5)+(48.*H1+32.*H2+32.*H3)*C(ID,N6))
      REL(5)     = REL(5)             -B*((4.*H1+12.*H2+12.*H3)*C(ID,N1)*(-1)
     1-(4.*H1-12.*H2+8.*H3)*C(ID,N2)-(4.*H1+8.*H2-12.*H3)*C(ID,N3)
     2+(32.*H1+48.*H2+32.*H3)*C(ID,N4)+(32.*H1+96.*H2+96.*H3)*
     3C(ID,N5)+(32.*H1+32.*H2+48.*H3)*C(ID,N6))
      REL(6)     = REL(6)                -B*((12.*H1-4.*H2-8.*H3)*C(ID,N1)
     1-(12.*H1+4.*H2+12.*H3)*C(ID,N2)-(8.*H1+4.*H2-12.*H3)*C(ID,N3)
     2+(48.*H1+32.*H2+32.*H3)*C(ID,N4)+(32.*H1+32.*H2+48.*H3)*
     3C(ID,N5)+(96.*H1+32.*H2+96.*H3)*C(ID,N6))
C
C         IF(M.EQ.77) WRITE(6,999) ID,(REL(I),I=1,6)
C     POINT SOURCE CONTRIBUTION
C
      B          = 1.0
```

```
C
      REL(1)   = REL(1)               +B*((30.*H1+6.*H2+6.*H3)*QPNT(ID,N1)
     1-(4.*H1+4.*H2-H3)*QPNT(ID,N2)     -(4.*H1-H2+4.*H3)*QPNT(ID,N3)
     2+(12.*H1-8.*H2-4.*H3)*QPNT(ID,N4)-(4.*H1+12.*H2+12.*H3)*
     3QPNT(ID,N5)+(12.*H1-4.*H2-8.*H3)*QPNT(ID,N6))
      REL(2)   = REL(2)               +B*((4.*H1+4.*H2-H3)*QPNT(ID,N1)*(-1)
     1+(6.*H1+30.*H2+6.*H3)*QPNT(ID,N2)+(H1-4.*H2-4.*H3)*QPNT(ID,N3)
     2-(8.*H1-12.*H2+4.*H3)*QPNT(ID,N4)-(4.*H1-12.*H2+8.*H3)*QPNT(ID,N5)
     3-(12.*H1+4.*H2+12.*H3)*QPNT(ID,N6))
      REL(3)   = REL(3)               +B*((4.*H1-H2+4.*H3)*QPNT(ID,N1)*(-1)
     1+(H1-4.*H2-4.*H3)*QPNT(ID,N2)     +(6.*H1+6.*H2+30.*H3)*QPNT(ID,N3)
     2-(12.*H1+12.*H2+4.*H3)*QPNT(ID,N4)-(4.*H1+8.*H2-12.*H3)*
     3QPNT(ID,N5)-(8.*H1+4.*H2-12.*H3)*QPNT(ID,N6))
      REL(4)   = REL(4)               +B*((12.*H1-8.*H2-4.*H3)*QPNT(ID,N1)
     1-(8.*H1-12.*H2+4.*H3)*QPNT(ID,N2)-(12.*H1+12.*H2+4.*H3)*
     2QPNT(ID,N3)+(96.*H1+96.*H2+32.*H3)*QPNT(ID,N4)+(32.*H1+48.*H2+32.*
     3H3)*QPNT(ID,N5)+(48.*H1+32.*H2+32.*H3)*QPNT(ID,N6))
      REL(5)   = REL(5)    +B*((4.*H1+12.*H2+12.*H3)*QPNT(ID,N1)*(-1)
     1-(4.*H1-12.*H2+8.*H3)*QPNT(ID,N2)-(4.*H1+8.*H2-12.*H3)*QPNT(ID,N3)
     2+(32.*H1+48.*H2+32.*H3)*QPNT(ID,N4)+(32.*H1+96.*H2+96.*H3)*
     3QPNT(ID,N5)+(32.*H1+32.*H2+48.*H3)*QPNT(ID,N6))
      REL(6)   = REL(6)               +B*((12.*H1-4.*H2-8.*H3)*QPNT(ID,N1)
     1-(12.*H1+4.*H2+12.*H3)*QPNT(ID,N2)-(8.*H1+4.*H2-12.*H3)*
     2QPNT(ID,N3)+(48.*H1+32.*H2+32.*H3)*QPNT(ID,N4)+(32.*H1+32.*H2+48.
     3*H3)*QPNT(ID,N5)+(96.*H1+32.*H2+96.*H3)*QPNT(ID,N6))
C
C        IF(M.EQ.77) WRITE(6,999) ID,(REL(I),I=1,6)
      RETURN
      END
      SUBROUTINE PMATRX (M,AREA)
C
C     THIS SUBROUTINE CONSTRUCTS THE SYMMETRIC ELEMENTAL MATRIX "PEL"
C     FOR THE M-TH TRIANGULAR ELEMENT.
C
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
      COMMON /BLK10/PEL(6,6),SEL(6,6),REL(6)
C
      N1         = NOD(M,1)
      N2         = NOD(M,2)
      N3         = NOD(M,3)
      H1         = H(N1)
      H2         = H(N2)
      H3         = H(N3)
      A          = AREA/1260.
C
      PEL(1,1) = A*(30.*H1+6.*H2+6.*H3)
      PEL(1,2) = -A*(4.*H1+4.*H2-H3)
      PEL(1,3) = -A*(4.*H1-H2+4.*H3)
      PEL(1,4) = A*(12.*H1-8.*H2-4.*H3)
      PEL(1,5) = -A*(4.*H1+12.*H2+12.*H3)
      PEL(1,6) = A*(12.*H1-4.*H2-8.*H3)
      PEL(2,2) = A*(6.*H1+30.*H2+6.*H3)
      PEL(2,3) = A*(H1-4.*H2-4.*H3)
      PEL(2,4) = -A*(8.*H1-12.*H2+4.*H3)
      PEL(2,5) = -A*(4.*H1-12.*H2+8.*H3)
      PEL(2,6) = -A*(12.*H1+4.*H2+12.*H3)
      PEL(3,3) = A*(6.*H1+6.*H2+30.*H3)
      PEL(3,4) = -A*(12.*H1+12.*H2+4.*H3)
      PEL(3,5) = -A*(4.*H1+8.*H2-12.*H3)
      PEL(3,6) = -A*(8.*H1+4.*H2-12.*H3)
      PEL(4,4) = A*(96.*H1+96.*H2+32.*H3)
```

```
      PEL(4,5) = A*(32.*H1+48.*H2+32.*H3)
      PEL(4,6) = A*(48.*H1+32.*H2+32.*H3)
      PEL(5,5) = A*(32.*H1+96.*H2+96.*H3)
      PEL(5,6) = A*(32.*H1+32.*H2+48.*H3)
      PEL(6,6) = A*(96.*H1+32.*H2+96.*H3)
C
      DO 100 I=1,5
      II=I+1
      DO 100 J=II,6
      PEL(J,I)=PEL(I,J)
100   CONTINUE
C
      RETURN
      END
      SUBROUTINE RMATRX (M,AREA,ID,INO)
C
C     THIS SUBROUTINE CONSTRUCTS THE ELEMENTAL LOAD VECTOR (REL) FOR
C     THE M-TH TRIANGULAR ELEMENT.
C
      COMMON /BLK6/ VX(240),VY(240),H(240),STRESS(3)
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
      COMMON /BLK8/KBC,LBC,MBC(7,120),NBC(7,120),DBC(7,120),BC(7,120),
     1 KODE(10)
      COMMON /BLK9/DX(100),DY(100),EX(100),EY(100),ALFA(100),BETA(100)
     9            ,HS(100)
      COMMON /BLK10/PEL(6,6),SEL(6,6),REL(6)
      DIMENSION Z(6)
C
C     SOURCE TERM CONTRIBUTION
C
      N1 = NOD(M,1)
      N2 = NOD(M,2)
      N3 = NOD(M,3)
      H1 = H(N1)
      H2 = H(N2)
      H3 = H(N3)
C
      IF (INO.GT.0) GO TO 105
      DO 100 I=1,6
      REL(I)=0.0
      Z(I)=BETA(M)*AREA/3.0
      IF (I.LE.3) Z(I)=0.0
      Z(I) = Z(I) * (H1+H2+H3)/3.0
100   CONTINUE
C
C     INSERT THE DERIVATIVE BOUNDARY CONDITIONS
C
105   CONTINUE
      IF (KBC.LE.0) GO TO 120
      DO 110 K=4,6
      NSIDE=NOD(M,K)
      DO 110 I=1,KBC
      IF (NSIDE.NE.MBC(ID,I)) GO TO 110
      FBC1=0.
      FBC2=0.
      FBC3=0.
      IF (K.EQ.5) FBC1=DBC(ID,I)
      IF (K.EQ.4) FBC3=DBC(ID,I)
      N1=NOD(M,1)
      N2=NOD(M,2)
      IF (K.EQ.6) FBC2=DBC(ID,I)
      N3=NOD(M,3)
```

```
      S1=SQRT((X(N3)-X(N2))**2+(Y(N3)-Y(N2))**2)
      S2=SQRT((X(N3)-X(N1))**2+(Y(N3)-Y(N1))**2)
      S3=SQRT((X(N2)-X(N1))**2+(Y(N2)-Y(N1))**2)
C
      REL(1)=(FBC2*S2+FBC3*S3)/6.*(H1+H2+H3)/3.+REL(1)
      REL(2)=(FBC3*S3+FBC1*S1)/6.*(H1+H2+H3)/3.+REL(2)
      REL(3)=(FBC1*S1+FBC2*S2)/6.*(H1+H2+H3)/3.+REL(3)
      REL(4)=2.*FBC3*S3/3.*(H1+H2+H3)/3.+REL(4)
      REL(5)=2.*FBC1*S1/3.*(H1+H2+H3)/3.+REL(5)
      REL(6)=2.*FBC2*S2/3.*(H1+H2+H3)/3.+REL(6)
110   CONTINUE
C
120   CONTINUE
      DO 130 I=1,6
      REL(I)=REL(I)+Z(I)
130   CONTINUE
C
C
      RETURN
      END
      SUBROUTINE RVEL (AT,RAT,NCNDS,VFREQ)
C
C**   THIS SUBROUTINE READS DISCHARGE AND DEPTH DATA FROM THE DIRECT
C**   ACCESS FILES GENERATED BY CAFE. LOGICAL UNITS LUQ AND LUH MUST BE
C**   CONNECTED TO THE APPROPRIATE FILES THRU JOB CONTROL.
C
      COMMON /BLK6/ VX(240), VY(240), H(240), STRESS(3)
      COMMON /FILE1/DUM(2000)
      LEVEL  2,DUM
C
      DIMENSION NOD(135)
C
      IF (AT .GT. RAT) GO TO 200
C
C**   THE FOLLOWING SEGMENT SPECIFIES THE CORRELATION BETWEEN THE CAFE
C**   AND FETRA NODE NUMBERS. READ IN THE FETRA NODE NUMBERS THAT
C**   CORRESPOND TO THE CAFE NODES. THESE VALUES MUST BE READ IN AN
C**   ASCENDING CAFE NODE NUMBER ORDER.
C
C**   AMAX IS THE TIME IN DAYS WHERE THE VELOCITIES WILL
C**   BE STARTED OVER.
C
      READ (5,4000) AMAX
4000  FORMAT (F10.0)
      READ (5,5000) (NOD(I),I=1,NCNDS)
5000  FORMAT (16I5)
C
C**   THE VALUES IN THE "DEFINE FILE" STATEMENT ARE DEPENDENT
C**   UPON THE WAY THE FILES WERE CREATED IN THE GENERATING
C**   PROGRAM.
C
      LUQ = 7
      LUH = 8
C
      NREC = 1
C     DEFINE FILE 7 (338,912,U,NREC)
C     DEFINE FILE 8 (338,456,U,NREC)
C     DEFINE FILE 7 (50,520,U,NREC)
C     DEFINE FILE 8 (50,260,U,NREC)
C
 200  A = AT
      IF (A .GT. AMAX) A = AMOD(A,AMAX)
```

```
      AII = (A + VFREQ / 2.) / VFREQ + 1.                               RVEL      42
      II = AII                                                          RVEL      43
      IIVX=(II - 1) * NCNDS * 2 + 1                                     FETFIX5    2
      IIH=(II - 1) * NCNDS + 1                                          FETFIX5    3
C     READ (LUQ'II) (VX(NOD(I)), VY(NOD(I)), I=1,NCNDS)                 RVEL      44
C     READ (LUH'II) (H(NOD(I)),I=1,NCNDS)                               RVEL      45
C                                                                       RVEL      46
C**   SINCE THE DISCHARGE FILE GENERATED BY CAFE ACTUALLY CONTAINS      RVEL      47
C**   DISCHARGES PER UNIT WIDTH, THESE VALUES MUST BE CONVERTED TO      RVEL      48
C**   VELOCITIES. ALSO MUST CONVERT SECONDS TO DAYS.                    RVEL      49
      NCNDS2=NCNDS * 2                                                  FETFIX5    4
      CALL RDABSF (LUQ,DUM(1),NCNDS2,IIVX)                              FETFIX5    5
      IF(UNIT(LUQ)) 220,210,210                                         FETFIX5    6
C                                                                       FETFIX5    7
210   CONTINUE                                                          FETFIX5    8
C                                                                       FETFIX5    9
      WRITE(6,20010)                                                    FETFIX5   10
20010 FORMAT(5X,"DIRECT ACCESS ERROR ON VELOCITY FILE")                 FETFIX5   11
      STOP                                                              FETFIX5   12
C                                                                       FETFIX5   13
220   CONTINUE                                                          FETFIX5   14
C                                                                       FETFIX5   15
      ICNT=1                                                            FETFIX5   16
      DO 225 I=1,NCNDS                                                  FETFIX5   17
      VX(NOD(I))=DUM(ICNT)                                              FETFIX5   18
      ICNT=ICNT + 1                                                     FETFIX5   19
      VY(NOD(I))=DUM(ICNT)                                              FETFIX5   20
      ICNT=ICNT + 1                                                     FETFIX5   21
225   CONTINUE                                                          FETFIX5   22
C                                                                       FETFIX5   23
      CALL RDABSF (LUH,DUM(1),NCNDS,IIH)                                FETFIX5   24
      IF(UNIT(LUH))240,230,230                                          FETFIX5   25
C                                                                       FETFIX5   26
230   CONTINUE                                                          FETFIX5   27
C                                                                       FETFIX5   28
      WRITE (6,20020)                                                   FETFIX5   29
20020 FORMAT(5X,"DIRECT ACCESS ERROR ON DEPTH FILE")                    FETFIX5   30
      STOP                                                              FETFIX5   31
C                                                                       FETFIX5   32
240   CONTINUE                                                          FETFIX5   33
C                                                                       FETFIX5   34
      ICNT=1                                                            FETFIX5   35
      DO 245 I=1,NCNDS                                                  FETFIX5   36
      H(NOD(I))=DUM(ICNT)                                               FETFIX5   37
      ICNT=ICNT + 1                                                     FETFIX5   38
245   CONTINUE                                                          FETFIX5   39
C                                                                       FETFIX5   40
C                                                                       RVEL      50
      DO 300 I=1,NCNDS                                                  RVEL      51
      VX(NOD(I)) = VX(NOD(I)) / H(NOD(I)) * 3600. * 24.                 RVEL      52
      VY(NOD(I)) = VY(NOD(I)) / H(NOD(I)) * 3600. * 24.                 RVEL      53
 300  CONTINUE                                                          RVEL      54
      RETURN                                                            RVEL      58
      END                                                               RVEL      59
      SUBROUTINE RWAVE (AT,WFREQ)                                       RWAVE      2
C                                                                       RWAVE      3
C**   THIS SUBROUTINE DETERMINES WHICH FILE TO READ THE WAVE            RWAVE      4
C**   CHARACTERISTICS FROM, BASED ON THE WIND VELOCITY AND              RWAVE      5
C**   DIRECTION READ IN SUBROUTINE RWIND.                               RWAVE      6
C                                                                       RWAVE      7
C     CHARACTER*5 ZTYP                                                  FETFIX1   12
C                                                                       RWAVE      9
```

```
      COMMON /WIND/ WVEL(700), WANG(700)
      COMMON/WAVE/ NODE(240),  NC(135), A(135,10), K(135,10)
     1             , W(135,10), HB(135), ALPHAC(135), WAVE, N, D, VIS
     2          , DPTHS(135)
      COMMON /ZTYPE/ ZTYP(135)
      COMMON/FILE/ DUMM(2000),NVAR(2,100)
      DIMENSION IDUMM(2000)
      LEVEL 2, IDUMM,DUMM,NVAR
      EQUIVALENCE (IDUMM(1),DUMM(1))
C
C
      LOGICAL WAVE
C
      REAL K
C
      DIMENSION NDS(135), HINFS(20,18), DEPTH(20,18), ALFS(20,18),
     1 HBS(20,18), WLS(20,18)
      CALL RDABSF (3,IDUMM(1),2,1)
      IF(UNIT(3)) 10,5,5
5     CONTINUE
C
      WRITE(6,20010)
20010 FORMAT(" DIRECT ACCESS ERROR IN RWAVE")
      STOP
C
10    CONTINUE
C
      NREC=IDUMM(1)
      NVLOC=IDUMM(2)
C
      CALL RDABSF(3,NVAR(1,1),NREC * 2,NVLOC)
C
      IF(UNIT(3)) 12,5,5
C
12    CONTINUE
C
C
      ACOUNT = (AT + WFREQ / 2.) / WFREQ + 1.
      NCOUNT = ACOUNT
      ANG = WANG(NCOUNT)
      VEL = WVEL(NCOUNT)
C
C**   CHECK IF THE WIND IS BLOWING FROM OVERLAND. IF IT IS, ASSUME
C**   THERE ARE NO WAVES.
C
      IF((ANG .LT. 11.25).OR.(ANG .GT. 191.25)) GO TO 1000
      IF (ANG .GE. 33.75) GO TO 20
C
C**   NNE WAVES.
C
C     OPEN (UNIT=3,NAME="NNEWAVZON.DAT",TYPE="OLD",ACCESS="DIRECT")
C     OPEN (UNIT=4,NAME="NNESURF.DT2",TYPE="OLD",ACCESS="DIRECT")
      GO TO 90
   20 IF (ANG .GE. 56.25) GO TO 30
C
C**   NE WAVES.
C
C     OPEN (UNIT=3,NAME="NEWAVZON.DAT",TYPE="OLD",ACCESS="DIRECT")
C     OPEN (UNIT=4,NAME="NESURF.DT2",TYPE="OLD",ACCESS="DIRECT")
      GO TO 90
   30 IF (ANG .GE. 78.75) GO TO 40
C
```

```
C**    ENE WAVES.
C
C      OPEN (UNIT=3,NAME="ENEWAVZON.DAT",TYPE="OLD",ACCESS="DIRECT")
C      OPEN (UNIT=4,NAME="ENESURF.DT2",TYPE="OLD",ACCESS="DIRECT")
       GO TO 90
   40 IF (ANG .GE. 101.25) GO TO 50
C
C**    E WAVES.
C
C      OPEN (UNIT=3,NAME="EWAVZON.DAT",TYPE="OLD",ACCESS="DIRECT")
C      OPEN (UNIT=4,NAME="ESURF.DT2",TYPE="OLD",ACCESS="DIRECT")
       GO TO 90
   50 IF (ANG .GE. 123.75) GO TO 60
C
C**    ESE WAVES.
C
C      OPEN (UNIT=3,NAME="ESEWAVZON.DAT",TYPE="OLD",ACCESS="DIRECT")
C      OPEN (UNIT=4,NAME="ESESURF.DT2",TYPE="OLD",ACCESS="DIRECT")
       GO TO 90
   60 IF (ANG .GE. 146.25) GO TO 70
C
C**    SE WAVES.
C
C      OPEN (UNIT=3,NAME="SEWAVZON.DAT",TYPE="OLD",ACCESS="DIRECT")
C      OPEN (UNIT=4,NAME="SESURF.DT2",TYPE="OLD",ACCESS="DIRECT")
       GO TO 90
   70 IF (ANG .GE. 168.75) GO TO 80
C
C**    SSE WAVES.
C
C      OPEN (UNIT=3,NAME="SSEWAVZON.DAT",TYPE="OLD",ACCESS="DIRECT")
C      OPEN (UNIT=4,NAME="SSESURF.DT2",TYPE="OLD",ACCESS="DIRECT")
       GO TO 90
C
C**    S WAVES.
C
  80   CONTINUE
C      OPEN (UNIT=3,NAME="SWAVZON.DAT",TYPE="OLD",ACCESS="DIRECT")
C      OPEN (UNIT=4,NAME="SSURF.DT2",TYPE="OLD",ACCESS="DIRECT")
C
C**    THE FOLLOWING 2 EQUATIONS FOR CALCULATING WAVE PERIOD AND
C**    DEEPWATER WAVE HEIGHT ARE BASED ON THE METHOD DEVELOPED BY
C**    HASSELMANN ET AL, 1976 AS USED BY THE U. S. ARMY CORPS OF
C**    ENGINEERS. THE EQUATIONS ARE FOR FULLY DEVELOPED WAVES.
C
   90 PER = VEL/1.372
      HINF = .02478*VEL**2
C
C**    IF THE DEEPWATER WAVE HEIGHT IS LESS THAN .5 FT.
C**    OR THE WAVE PERIOD IS LESS THAN 2. SEC.
C**    ASSUME THERE ARE NO WAVES.
C
      IF (HINF*3.2808 .LT. .5) GO TO 1000
      IF (PER .LT. 2.) GO TO 1000
      IF (PER .GE. 4.) GO TO 120
      PER = 3.
      GO TO 160
  120 IF (PER .GE. 6.) GO TO 130
      PER = 5.
      GO TO 160
  130 IF (PER .GE. 8.) GO TO 140
      PER = 7.
```

```
      GO TO 160                                                         RWAVE    110
  140 IF (PER .GE. 10.) GO TO 150                                       RWAVE    111
      PER = 9.                                                          RWAVE    112
      GO TO 160                                                         RWAVE    113
  150 PER = 11.                                                         RWAVE    114
  160 II = (PER-1)/2                                                    RWAVE    115
C     READ (3'II) NV, (NDS(I), A(I,1), K(I,1),I=1,NV)                   RWAVE    116
      NV=NVAR(1,II)                                                     FETFIX5   65
      NVLOC=NVAR(2,II)                                                  FETFIX5   66
      CALL RDABSF(3,IDUMM(1),NV * 3,NVLOC)                              FETFIX5   67
      IF(UNIT(3)) 165,162,162                                           FETFIX5   68
C                                                                       FETFIX5   69
162   CONTINUE                                                          FETFIX5   70
C                                                                       FETFIX5   71
      WRITE(6,20010)                                                    FETFIX5   72
      STOP                                                              FETFIX5   73
C                                                                       FETFIX5   74
165   CONTINUE                                                          FETFIX5   75
C                                                                       FETFIX5   76
      ICNT=1                                                            FETFIX5   77
      DO 170 I=1,NV                                                     FETFIX5   78
      NDS(I)=IDUMM(ICNT)                                                FETFIX5   79
      ICNT=ICNT + 1                                                     FETFIX5   80
      A(I,1)=DUMM(ICNT)                                                 FETFIX5   81
      ICNT=ICNT + 1                                                     FETFIX5   82
      K(I,1)=DUMM(ICNT)                                                 FETFIX5   83
      ICNT=ICNT + 1                                                     FETFIX5   84
170   CONTINUE                                                          FETFIX5   85
C                                                                       FETFIX5   86
      PI = ACOS(-1.)                                                    RWAVE    117
      DO 300 I=1,NV                                                     RWAVE    118
      NC(I) = 1                                                         RWAVE    119
      NODE(NDS(I)) = I                                                  RWAVE    120
      A(I,1) = A(I,1)*HINF/2.                                           RWAVE    121
      K(I,1) = 2.*PI/K(I,1)*3.2808                                      RWAVE    122
      W(I,1) = 2.*PI/PER                                                RWAVE    123
      ZTYP(I) = "WAVE,"                                                 RWAVE    124
C     WRITE (6,4000) I, A(I,1), K(I,1), W(I,1)                          RWAVE    125
C4000 FORMAT (" I, A(I,1), K(I,1), W(I,1) = ",I5,1P3E12.4)              RWAVE    126
  300 CONTINUE                                                          RWAVE    127
C                                                                       RWAVE    128
C**   THE FOLLOWING SEGMENT READS THE SURF ZONE DATA.                   RWAVE    129
C**   IT ASSUMES THAT THE SURF ZONE DATA WAS GENERATED                  RWAVE    130
C**   FOR DEEPWATER WAVE HEIGHTS FROM 1 TO 18 FEET IN                   RWAVE    131
C**   1 FOOT INCREMENTS.                                                RWAVE    132
C                                                                       RWAVE    133
      NUMNOD = 18                                                       RWAVE    134
      NUMHT = 18                                                        RWAVE    135
C     READ (4'II) (NDS(I),(HINFS(I,J),DEPTH(I,J),HBS(I,J),ALFS(I,J),    RWAVE    136
C    1 WLS(I,J),J=1,NUMHT),I=1,NUMNOD)                                  FETFIX1   13
C                                      $2                               FETFIX5   87
      NWDS=5 * NUMHT * NUMNOD + NUMNOD                                  FETFIX5   88
      NLOC=(II - 1) * NWDS + 1                                          FETFIX5   89
      CALL RDABSF (4,IDUMM(1),NWDS,NLOC)                                FETFIX5   90
      IF(UNIT(4)) 305,302,302                                           FETFIX5   91
C                                                                       FETFIX5   92
302   CONTINUE                                                          FETFIX5   93
C                                                                       FETFIX5   94
      WRITE(6,20010)                                                    FETFIX5   95
      STOP                                                              FETFIX5   96
C                                                                       FETFIX5   97
305   CONTINUE                                                          FETFIX5   98
```

```
C
      ICNT=1
      DO 320 I=1,NUMNOD
      NDS(I)=IDUMM(ICNT)
      ICNT=ICNT + 1
      DO 310 J=1,NUMHT
      HINFS(I,J)=DUMM(ICNT)
      ICNT=ICNT + 1
      DEPTH(I,J)=DUMM(ICNT)
      ICNT=ICNT + 1
      HBS(I,J)=DUMM(ICNT)
      ICNT=ICNT + 1
      ALFS(ICNT)=DUMM(ICNT)
      ICNT=ICNT + 1
      WLS(ICNT)=DUMM(ICNT)
      ICNT=ICNT + 1
310   CONTINUE
C
320   CONTINUE
C
      L = NV + 1
      DO 500 I=1,NUMNOD
      NC(L) = 1
      NODE(NDS(I)) = L
      ZTYP(L) = "SURF,"
      DO 400 J=1,NUMHT
      DELH = HINFS(I,J) - HINF * 3.2808
 400  IF (ABS(DELH) .LE. .5) GO TO 350
      GO TO 410
 350  ALPHAC(L) = ALFS(I,J)
      HB(L) = HBS(I,J) / 3.2808
      K(L,1) = 2. * PI / WLS(I,J) * 3.2808
      DPTHS(L) = DEPTH(I,J) / 3.2808
      GO TO 490
 410  CONTINUE
C
C**       IF NO SURF DATA IS APPLICABLE TO THE NODE
C**       ASSUME THE WATER IS CALM.
C
      ALPHAC(L)=0.
      HB(L)=0.
      K(L,1)=9999.
      WRITE (6,6000) NDS(I), HINF
6000  FORMAT (1X,"SURF ZONE DATA NOT AVAILABLE FOR NODE ",I5/
     1 1X,"DEEPWATER WAVE HEIGHT = ",1PE10.3)
      WRITE(6,6010)
6010  FORMAT(1X,"THE WATER AT THIS NODE IS ASSUMED CALM.")
 490  CONTINUE
C     WRITE (6,4001) L, ALPHAC(L), HB(L), K(L,1), DPTHS(L)
C4001 FORMAT (" L, ALPHAC(L), HB(L), K(L,1), DPTHS(L) = ",I5,
C    1 1P4E12.4)
      L =L + 1
 500  CONTINUE
      GO TO 2000
 1000 DO 1010 I=1,409
 1010 NODE(I) = 0
      WRITE (6,2010) AT,ANG,VEL
 2010 FORMAT (/" AT TIME =",F8.4," DAYS, THE WAVES ARE TOO SMALL"
     1 " TO CAUSE APPRECIABLE TRANSPORT."/" WIND DIRECTION =",
     2 F8.4," DEGREES FROM TRUE NORTH."/" WIND VELOCITY =" F8.4,
     3 " M/SEC.")
 2000 NCOUNT = NCOUNT+1
```

```
      CALL CLOSE(3)                                                     RWAVE     180
      CALL CLOSE(4)                                                     RWAVE     181
      RETURN                                                            RWAVE     182
      END                                                               RWAVE     183
      SUBROUTINE RWIND (WFREQ,LW)                                       RWIND       2
C                                                                       RWIND       3
C**   THIS SUBROUTINE READS THE WIND DATA REQUIRED FOR COMPUTATION      RWIND       4
C**   OF THE WAVE CHARACTERISTICS.                                      RWIND       5
C                                                                       RWIND       6
      COMMON /WIND/ WVEL(700), WANG(700)                                RWIND       7
C                                                                       RWIND       8
C**   LW      = NUMBER OF WIND DATA POINTS TO BE INPUT. THE FIRST WIND  RWIND       9
C**             DATA POINT WILL BE USED AT TIME=0. DAYS.                RWIND      10
C**   WFREQ  = TIME INTERVAL IN DAYS BETWEEN EACH WIND DATA POINT.      RWIND      11
C**   WANG   = DIRECTION FROM WHICH WIND IS BLOWING, DEGREES FROM       RWIND      12
C**             TRUE NORTH MEASURED CW.                                 RWIND      13
C**   WVEL   = WIND VELOCITY, M/SEC.                                    RWIND      14
C                                                                       RWIND      15
      READ (5,5000) LW,WFREQ                                            RWIND      16
      READ (5,5010) ((WVEL(I), WANG(I)),I=1,LW)                         RWIND      17
C                                                                       RWIND      18
C      WRITE (6,6000) (WVEL(I),WANG(I),I=1,LW)                          RWIND      19
C6000  FORMAT (" ******* WVEL(I),WANG(I) = ",8F10.2)                    RWIND      20
      RETURN                                                            RWIND      21
 5000 FORMAT (I5,F10.0)                                                 RWIND      22
 5010 FORMAT (8E10.2)                                                   RWIND      23
      END                                                               RWIND      24
      SUBROUTINE SAND (M,ID,T,AREA)                                     SAND        2
C                                                                       SAND        3
C     THIS SUBROUTINE COMPUTES CAPACITY OF SEDIMENT LOAD                SAND        4
C     AND THEN SUBTRACT IT FROM THE ACTUAL LOAD TO OBTAIN THE AMOUNT OF SAND        5
C     SEDIMENT RESUSPENSION OR DEPOSITION                               SAND        6
C                                                                       SAND        7
C     CHARACTER*5 ZTYP                                                  FETFIX1    14
      REAL K                                                            SAND        9
      COMMON NDS                                                        SAND       10
      COMMON /BLK1/CT(240),C(9,240)                                     SAND       11
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)                     SAND       12
      COMMON /BLK7/NOD(240,6),X(240),Y(240)                             SAND       13
      COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)            SAND       14
      COMMON /BLK14/GBA(100,10),GBB(100,10),GBC(100,10),GBD(100,10),    SAND       15
     1              GBE(100,10),GBF(100,10),GBG(100,10),POR             SAND       16
      COMMON /BLK15/ILAYR(100,3),XYS0(100),BDIV(100),NBED(100),BED(100), SAND      17
     1             RHOSED(3),XNT(100,3),RSAV1(100),RSAV2(100),RSAV3(100) SAND      18
      COMMON/WAVE/ NODE(240),  NC(135), A(135,10), K(135,10),           SAND       19
     1             WW(135,10), HB(135), ALPHAC(135), WAVE, N, D         SAND       20
     2             , VIS, DPTHS(135)                                    SAND       21
      COMMON /ZTYPE/ ZTYP(135)                                          SAND       22
      LOGICAL WAVE                                                      SAND       23
      DIMENSION QS(3),QSA(3)                                            SAND       24
      DIMENSION QSS(3)                                                  FETFIX8     6
      DIMENSION NCHECK(240), QCHECK(240)                                SAND       25
      DATA EPSI /1.E-10/                                                SAND       26
C                                                                       SAND       27
C**   NCHECK IS USED TO ENSURE THAT THE SEDIMENT CAPACITY SUBROUTINES   SAND       28
C**   (DUBOY, WAVSAN, AND SURFTR) ARE CALLED ONLY ONCE FOR EACH NODE    SAND       29
C**   AT A GIVEN TIME STEP. QCHECK STORES THE RESULT FROM EACH NODE.    SAND       30
C                                                                       SAND       31
      PI = ACOS(-1.0)                                                   SAND       32
      IF(M .NE. 1) GO TO 3                                              SAND       33
      DO 2 I=1,NDS                                                      SAND       34
      NCHECK(I)=0                                                       SAND       35
```

```
    2 CONTINUE
    3 SD(ID,M)=0.0
      SR(ID,M)=0.0
      RSAV3(M)=0.
      RS=0.0
      ILAYR(M,ID)=0
      TOTAL = GBA(M,NBED(M))/RHOSED(1)+GBB(M,NBED(M))/RHOSED(2)+
     1 GBC(M,NBED(M))/RHOSED(3)
      XDTOP=(1.-POR)/TOTAL
      XNT(M,ID)=XYSO(M)*GBC(M,NBED(M))*XDTOP
      DO 400 I=1,3
      NI=NOD(M,I)
      IF(NCHECK(NI) .EQ. 0) GO TO 5
      QS(I)=QCHECK(NI)
      GO TO 400
    5 N=NODE(NI)
      QS(I) = 0.0
      QSS(I) = 0.0
      D=D50(ID,M)
      UEXT=SQRT(VX(NI)**2+VY(NI)**2)
      IF(N .GT. 0) GO TO 10
      CALL DUBOY(ID,M,I,QSS(I))
   10 IF(ZTYP(N) .EQ. "WAVE,") CALL WAVSAN(ID,H(NI),UEXT,QS(I))
C     IF(ZTYP(N) .EQ. "SURF,".AND.HB(N).GT.1.0E-10)
C    1 CALL SURFTR(ID,H(NI),UEXT,QS(I))
       IF(ZTYP(N).EQ."SURF,") CALL SURFTR(ID,H(NI),UEXT,QS(I))
  399 CONTINUE
      NCHECK(NI)=1
      QCHECK(NI) = QS(I)+QSS(I)
  400 CONTINUE
      DCTH=0.
      S=0.
      SN=0.
      DO 100 I=1,3
      IF(I-2) 60,70,80
C
C**   N1 IS THE NODE AT WHICH THE SCOUR OR EROSION COEFFICIENT IS
C**      BEING CALCULATED.
C**   N2 IS THE NEXT CORNER NODE COUNTER-CLOCKWISE.
C**   N3 IS THE SECOND CORNER NODE COUNTER-CLOCKWISE.
C
   60 N1=NOD(M,1)
      N2=NOD(M,2)
      N3=NOD(M,3)
      I1=1
      I2=2
      I3=3
      GO TO 90
   70 N1=NOD(M,2)
      N2=NOD(M,3)
      N3=NOD(M,1)
      I1=2
      I2=3
      I3=1
      GO TO 90
   80 N1=NOD(M,3)
      N2=NOD(M,1)
      N3=NOD(M,2)
      I1=3
      I2=1
      I3=2
   90 CONTINUE
```

```
      VM=SQRT(VX(N1)**2+VY(N1)**2)
      IF(VM .EQ. 0.) GO TO 91
      CTH=QS(I)/VM
      GO TO 93
C
C**   IF VM IS ZERO, ASSUME THAT EVENTUALLY ALL THE SUSPENDED
C**   SEDIMENT WILL SETTLE OUT.
C
   91 CTH=0.
   93 CAC=C(ID,N1)*H(N1)
      DCTH=DCTH+(CTH-CAC)
      IF(VM .EQ. 0.) GO TO 100
      QSA(I)=C(ID,N1)*H(N1)*VM
      A1=Y(N2)-Y(N1)
      A2=X(N2)-X(N1)
      A3=Y(N3)-Y(N1)
      A4=X(N3)-X(N1)
C
C**   ALPHA1 IS THE ANGLE BETWEEN THE X-AXIS AND THE ELEMENT SIDE, N1
C**           TO N2.
C**   ALPHA2 IS THE ANGLE BETWEEN THE X-AXIS AND THE ELEMENT SIDE N1
C**           TO N3.
C**   BETA IS THE ANGLE BETWEEN THE X-AXIS AND THE VELOCITY VECTOR AT N1
C**   DEN1 IS THE DISTANCE BETWEEN N1 AND N2.
C**   DEN2 IS THE DISTANCE BETWEEN N1 AND N3.
C
      ALPHA1=ATAN2(A1,A2)
      IF(ALPHA1 .LT. 0.) ALPHA1=ALPHA1+2.*PI
      ALPHA2=ATAN2(A3,A4)
      IF(ALPHA2 .LT. 0.) ALPHA2=ALPHA2+2.*PI
      BETA=ATAN2(VY(N1),VX(N1))
      IF(BETA .LT. 0.) BETA=BETA+2.*PI
      DEN1 =SQRT((X(N1)-X(N2))**2+(Y(N1)-Y(N2))**2)
      DEN2 =SQRT((X(N1)-X(N3))**2+(Y(N1)-Y(N3))**2)
      ANGLE=BETA-ALPHA1
C
C**   CHECK IF THERE IS A VELOCITY COMPONENT TOWARDS N2.
C
      IF(COS(ANGLE) .LE. 0.) GO TO 92
      SN=SN+1.
      S=S+(QS(I2)-QSA(I1))*COS(ANGLE)/DEN1
   92 ANGLE =BETA-ALPHA2
C
C**   CHECK IF THERE IS A VELOCITY COMPONENT TOWARDS N3.
C
      IF(COS(ANGLE) .LE. 0.) GO TO 100
      SN=SN+1.
      S=S+(QS(I3)-QSA(I1))*COS(ANGLE)/DEN2
  100 CONTINUE
      S2=S
      IF(SN .GT. 0.) S2=S2/SN
      DCTH=DCTH/3.
C
C**   CHECK TO SEE IF THE TIME STEP IS TOO LARGE.
C**   S2*T INDICATES THE AMOUNT OF SEDIMENT THAT WILL BE SCOURED OR
C**   DEPOSITED DURING THE NEXT TIME STEP.
C**   DCTH INDICATES THE DIFFERENCE BETWEEN THE THEORETICAL AND ACTUAL
C**   SEDIMENT CAPACITY ON A PER UNIT HORIZONTAL AREA BASIS.
C
      IF(DCTH .EQ. 0.) GO TO 110
      RATIO=S2*T/DCTH
      IF(RATIO .LE. 1.) GO TO 110
```

```
      DT=DCTH/S2
      WRITE(6,6000) M, S2
      WRITE(6,6001) DCTH, DT
  110 CONTINUE
C
      IF ( S2 ) 130,305,150
C     DEPOSITION
130   ILAYR(M,ID)=-1
      SD(ID,M)=(-1)*S2
      N1 = NOD(M,1)
      N2 = NOD(M,2)
      N3 = NOD(M,3)
      N4 = NOD(M,4)
      N5 = NOD(M,5)
      N6 = NOD(M,6)
      H1 = H(N1)
      H2 = H(N2)
      H3 = H(N3)
      HAVG = (H1+H2+H3)/3.
      CAVG3 = (C(ID,N1)*H1+C(ID,N2)*H2+C(ID,N3)*H3+C(ID,N4)*(H1+H2)/2.
     1  +C(ID,N5)*(H2+H3)/2.+C(ID,N6)*(H1+H3)/2.)/(6.*HAVG)
      IF (CAVG3.LE.EPSI) GO TO 305
      CAVG7 = (C(ID+4,N1)*H1+C(ID+4,N2)*H2+C(ID+4,N3)*H3+C(ID+4,N4)*
     1  (H1+H2)/2.+C(ID+4,N5)*(H2+H3)/2.+C(ID+4,N6)*(H1+H3)/2.)/
     2  (6.*HAVG)
      RSAV3(M) = -SD(ID,M)*CAVG7/CAVG3
      GO TO 305
C
C     RESUSPENSION
C
150   SR(ID,M)=S2
C
C     CHECK THE AVAILABILITY OF SAND IN THE BED TO BE SCOURED
C     TO COMPUTE THE AVAILABILITY OF SAND IN BED LAYERS
C     A NUMBER OF BED LAYERS SCOURED IS PREDETERMINED BY OTHER SEDIMENT
C     (CLAY OR OTHER SAND)
      NB = NBED(M)
      SR(ID,M)=SR(ID,M)*T
      IF (SR(ID,M).GT.XNT(M,ID).AND.ILAYR(M,1).GT.0) GO TO 200
      SR(ID,M)=AMIN1(SR(ID,M),XNT(M,ID))
      XNT(M,ID)=XNT(M,ID)-SR(ID,M)
      RSAV3(M)=SR(ID,M)*GBF(M,NBED(M))
      GO TO 290
  200 ILAYR(M,ID)=ILAYR(M,ID)+1
      RSAV3(M)=RSAV3(M)+XNT(M,ID)*GBF(M,NB)
      SR(ID,M)=SR(ID,M)-XNT(M,ID)
      RS=RS+XNT(M,ID)
      NB=NBED(M)-ILAYR(M,ID)
      IF (NB.LE.0) GO TO 285
      TOTAL=GBA(M,NB)/RHOSED(1)+GBB(M,NB)/RHOSED(2)+GBC(M,NB)/RHOSED(3)
      XND=(1.-POR)/TOTAL
      XNT(M,ID)=BDIV(M)*GBC(M,NB)*XND
      IF(ILAYR(M,ID) .GE. ILAYR(M,1)) GO TO 280
      IF(SR(ID,M).GE.XNT(M,ID)) GO TO 200
280   RS=RS+AMIN1(SR(ID,M),XNT(M,ID))
      RSAV3(M)=RSAV3(M)+AMIN1(SR(ID,M),XNT(M,ID))*GBF(M,NB)
  285 XNT(M,ID)=XNT(M,ID)-AMIN1(SR(ID,M),XNT(M,ID))
      SR(ID,M)=RS
290   SR(ID,M)=SR(ID,M)/T
      RSAV3(M)=RSAV3(M)/T
  305 CONTINUE
      RETURN
```

```
 6000 FORMAT(1X,"***WARNING***",/,                                      SAND     222
     $1X,"THE COMPUTED DEPOSITION RATE FOR ELEMENT ",I3," IS ",1PE12.4, SAND     223
     $" KGF/DAY-M**2.")                                                 SAND     224
 6001 FORMAT(1X,"THE THEORETICAL CAPACITY DIFFERENCE IS ",1PE12.4," KGF/ SAND     225
     $M**2",/,                                                          SAND     226
     $1X,"THEREFORE THE TIME STEP SHOULD BE REDUCED TO A MAXIMUM OF ",  SAND     227
     $E12.4," DAY.")                                                    SAND     228
      END                                                               SAND     229
      SUBROUTINE SAND2 (M,ID,T,AREA)                                    SAND2      2
C                                                                       SAND2      3
C      THIS SUBROUTINE COMPUTES THE SEDIMENT LOAD CAPACITY              SAND2      4
C     AND THEN SUBTRACT IT FROM THE ACTUAL LOAD TO OBTAIN THE AMOUNT OF SAND2      5
C     SEDIMENT RESUSPENSION OR DEPOSITION                               SAND2      6
C     THIS ROUTINE DECIDES NO. OF BED LAYERS TO BE SCOURED              SAND2      7
C     SUBROUTINE CLAY OR THIS DECIDE NO. OF SCOURED BED LAYERS          SAND2      8
C                                                                       SAND2      9
C     CHARACTER*5 ZTYP                                                  FETFIX1   15
      REAL K                                                            SAND2     11
C                                                                       SAND2     12
      COMMON NDS                                                        SAND2     13
      COMMON /BLK1/CT(240),C(9,240)                                     SAND2     14
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)                     SAND2     15
      COMMON /BLK7/NOD(240,6),X(240),Y(240)                             SAND2     16
      COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)            SAND2     17
      COMMON /BLK14/GBA(100,10),GBB(100,10),GBC(100,10),GBD(100,10),    SAND2     18
     1             GBE(100,10),GBF(100,10),GBG(100,10),POR              SAND2     19
      COMMON /BLK15/ILAYR(100,3),XYS0(100),BDIV(100),NBED(100),BED(100), SAND2    20
     1             RHOSED(3),XNT(100,3),RSAV1(100),RSAV2(100),RSAV3(100) SAND2    21
      COMMON/WAVE/ NODE(240),  NC(135), A(135,10), K(135,10),           SAND2     22
     1             HW(135,10), HB(135), ALPHAC(135), WAVE, N, D         SAND2     23
     2             , VIS, DPTHS(135)                                    SAND2     24
      COMMON /ZTYPE/ ZTYP(135)                                          SAND2     25
      LOGICAL WAVE                                                      SAND2     26
      DIMENSION QS(3), QSA(3)                                           SAND2     27
      DIMENSION QSS(3)                                                  FETFIX8   10
      DIMENSION NCHECK(240), QCHECK(240)                                SAND2     28
      DATA EPSI /1.E-10/                                                SAND2     29
C                                                                       SAND2     30
C**   NCHECK IS USED TO ENSURE THAT THE SEDIMENT CAPACITY SUBROUTINES   SAND2     31
C**   (DUBOY,WAVSAN, AND SURFTR) ARE CALLED ONLY ONCE FOR EACH NODE     SAND2     32
C**   AT A GIVEN TIME STEP. QCHECK STORES THE RESULT FROM EACH NODE.    SAND2     33
C                                                                       SAND2     34
      PI=ACOS(-1.)                                                      SAND2     35
      IF(M .NE. 1) GO TO 3                                              SAND2     36
      DO 2 I=1,NDS                                                      SAND2     37
      NCHECK(I)=0                                                       SAND2     38
    2 CONTINUE                                                          SAND2     39
    3 SD(ID,M)=0.0                                                      SAND2     40
      SR(ID,M)=0.0                                                      SAND2     41
      RSAV1(M)=0.                                                       SAND2     42
      RS=0.0                                                            SAND2     43
      ILAYR(M,ID)=0                                                     SAND2     44
      TOTAL = GBA(M,NBED(M))/RHOSED(1)+GBB(M,NBED(M))/RHOSED(2)+        SAND2     45
     1 GBC(M,NBED(M))/RHOSED(3)                                         SAND2     46
      XDTOP=(1.-POR)/TOTAL                                              SAND2     47
      XNT(M,ID)=XYS0(M)*GBA(M,NBED(M))*XDTOP                            SAND2     48
      DO 400 I=1,3                                                      SAND2     49
      NI=NOD(M,I)                                                       SAND2     50
      IF(NCHECK(NI) .EQ. 0) GO TO 5                                     SAND2     51
      QS(I)=QCHECK(NI)                                                  SAND2     52
      GO TO 400                                                         SAND2     53
    5 N=NODE(NI)                                                        SAND2     54
```

```
      QS(I) = 0.0                                                       SAND2     55
      QSS(I) = 0.0                                                      FETFIX8   11
      D=D50(ID,M)                                                       SAND2     56
      UEXT=SQRT(VX(NI)**2+VY(NI)**2)                                    SAND2     57
      IF(N .GT. 0) GO TO 10                                             SAND2     58
      CALL DUBOY(ID,M,I,QSS(I))                                         FETFIX8   12
   10 IF(ZTYP(N) .EQ. "WAVE,") CALL WAVSAN(ID,H(NI),UEXT,QS(I))         SAND2     61
      IF(ZTYP(N).EQ."SURF," .AND. HB(N).GT.1.0E-10 .AND. DPTHS(N)       SAND2     62
     1 .GE.H(NI)) CALL SURFTR(ID,H(NI),UEXT,QS(I))                      SAND2     63
C     IF(ZTYP(N).EQ."SURF,")  CALL SURFTR(ID,H(NI),UEXT,QS(I))          SAND2     64
C      WRITE (6,6020) M,N,I,NI,QS(I),ZTYP(N)                            SAND2     65
C6020  FORMAT (" **** M, N, I, NI, QS(I), ZTYP(N) = ",4I5,1PE12.4,1X,A5) SAND2     66
  399 CONTINUE                                                          SAND2     67
      NCHECK(NI)=1                                                      SAND2     68
      QCHECK(NI) = QS(I) + QSS(I)                                       FETFIX8   13
  400 CONTINUE                                                          SAND2     70
      DCTH=0.                                                           SAND2     71
      S=0.                                                              SAND2     72
      SN=0.                                                             SAND2     73
      DO 100 I=1,3                                                      SAND2     74
      IF(I-2) 60,70,80                                                  SAND2     75
C                                                                       SAND2     76
C**   N1 IS THE NODE AT WHICH THE SCOUR OR EROSION COEFFICIENT IS       SAND2     77
C**      BEING CALCULATED.                                              SAND2     78
C**   N2 IS THE NEXT CORNER NODE COUNTER-CLOCKWISE.                     SAND2     79
C**   N3 IS THE SECOND CORNER NODE COUNTER-CLOCKWISE.                   SAND2     80
C                                                                       SAND2     81
   60 N1=NOD(M,1)                                                       SAND2     82
      N2=NOD(M,2)                                                       SAND2     83
      N3=NOD(M,3)                                                       SAND2     84
      I1=1                                                              SAND2     85
      I2=2                                                              SAND2     86
      I3=3                                                              SAND2     87
      GO TO 90                                                          SAND2     88
   70 N1=NOD(M,2)                                                       SAND2     89
      N2=NOD(M,3)                                                       SAND2     90
      N3=NOD(M,1)                                                       SAND2     91
      I1=2                                                              SAND2     92
      I2=3                                                              SAND2     93
      I3=1                                                              SAND2     94
      GO TO 90                                                          SAND2     95
   80 N1=NOD(M,3)                                                       SAND2     96
      N2=NOD(M,1)                                                       SAND2     97
      N3=NOD(M,2)                                                       SAND2     98
      I1=3                                                              SAND2     99
      I2=1                                                              SAND2    100
      I3=2                                                              SAND2    101
   90 CONTINUE                                                          SAND2    102
      VM=SQRT(VX(N1)**2+VY(N1)**2)                                      SAND2    103
      IF(VM .EQ. 0.) GO TO 91                                           SAND2    104
      CTH=QS(I)/VM                                                      SAND2    105
      GO TO 93                                                          SAND2    106
C                                                                       SAND2    107
C**   IF VM IS ZERO, ASSUME THAT EVENTUALLY ALL THE SUSPENDED           SAND2    108
C**   SEDIMENT WILL SETTLE OUT.                                         SAND2    109
C                                                                       SAND2    110
   91 CTH=0.                                                            SAND2    111
   93 CAC=C(ID,N1)*H(N1)                                                SAND2    112
      DCTH=DCTH+(CTH-CAC)                                               SAND2    113
      IF(VM .EQ. 0.) GO TO 100                                          SAND2    114
      QSA(I)=C(ID,N1)*H(N1)*VM                                          SAND2    115
      A1=Y(N2)-Y(N1)                                                    SAND2    116
```

```
      A2=X(N2)-X(N1)
      A3=Y(N3)-Y(N1)
      A4=X(N3)-X(N1)
C
C**   ALPHA1 IS THE ANGLE BETWEEN THE X-AXIS AND THE ELEMENT SIDE, N1
C**           TO N2.
C**   ALPHA2 IS THE ANGLE BETWEEN THE X-AXIS AND THE ELEMENT SIDE N1
C**           TO N3.
C**   BETA IS THE ANGLE BETWEEN THE X-AXIS AND THE VELOCITY VECTOR AT N1
C**   DEN1 IS THE DISTANCE BETWEEN N1 AND N2.
C**   DEN2 IS THE DISTANCE BETWEEN N1 AND N3.
C
      ALPHA1=ATAN2(A1,A2)
      IF(ALPHA1 .LT. 0.) ALPHA1=ALPHA1+2.*PI
      ALPHA2=ATAN2(A3,A4)
      IF(ALPHA2 .LT. 0.) ALPHA2=ALPHA2+2.*PI
      BETA=ATAN2(VY(N1),VX(N1))
      IF(BETA .LT. 0.) BETA=BETA+2.*PI
      DEN1 =SQRT((X(N1)-X(N2))**2+(Y(N1)-Y(N2))**2)
      DEN2 =SQRT((X(N1)-X(N3))**2+(Y(N1)-Y(N3))**2)
      ANGLE=BETA-ALPHA1
C      WRITE (6,7000) M,N1,N2,N3,ALPHA1,ALPHA2,BETA,ANGLE
C7000  FORMAT ("M,N1,N2,N3,ALPHA1,ALPHA2,BETA,ANGLE = "4I5,1P4E12.4)
C
C**   CHECK IF THERE IS A VELOCITY COMPONENT TOWARDS N2.
C
      IF(COS(ANGLE) .LE. 0.) GO TO 92
      SN=SN+1.
      S=S+(QS(I2)-QSA(I1))*COS(ANGLE)/DEN1
C     WRITE (6,1198) M,N1,N2,N3,I1,I2,S,QSA(I1),QS(I2)
C1198  FORMAT (" M,N1,N2,N3,I1,I2,S,QSA(I1),QS(I2) = ",6I5,1P3E12.4)
   92 ANGLE =BETA-ALPHA2
C
C**   CHECK IF THERE IS A VELOCITY COMPONENT TOWARDS N3.
C
      IF(COS(ANGLE) .LE. 0.) GO TO 100
      SN=SN+1.
      S=S+(QS(I3)-QSA(I1))*COS(ANGLE)/DEN2
C     WRITE (6,1199) M,N1,N2,N3,I1,I3,S,QSA(I1),QS(I3)
C1199  FORMAT (" M,N1,N2,N3,I1,I3,S,QSA(I1),QS(I3) = ",6I5,1P3E12.4)
  100 CONTINUE
      S2=S
      IF(SN .GT. 0.) S2=S2/SN
      DCTH=DCTH/3.
C
C**   CHECK TO SEE IF THE TIME STEP IS TOO LARGE.
C**   S2*T INDICATES THE AMOUNT OF SEDIMENT THAT WILL BE SCOURED OR
C**   DEPOSITED DURING THE NEXT TIME STEP.
C**   DCTH INDICATES THE DIFFERENCE BETWEEN THE THEORETICAL AND ACTUAL
C**   SEDIMENT CAPACITY ON A PER UNIT HORIZONTAL AREA BASIS.
C
      IF(DCTH .EQ. 0.) GO TO 110
      RATIO=S2*T/DCTH
      IF(RATIO .LE. 1.) GO TO 110
      DT=DCTH/S2
      S2=DCTH/DT/2.0
      WRITE(6,6000) M, S2
      WRITE(6,6001) DCTH, DT
  110 CONTINUE
C
      IF ( S2 ) 130,305,150
C     DEPOSITION
```

```
  130 ILAYR(M,ID)=-1
      SD(ID,M)=(-1)*S2
      N1 = NOD(M,1)
      N2 = NOD(M,2)
      N3 = NOD(M,3)
      N4 = NOD(M,4)
      N5 = NOD(M,5)
      N6 = NOD(M,6)
      H1 = H(N1)
      H2 = H(N2)
      H3 = H(N3)
      HAVG = (H1+H2+H3)/3.
      CAVG1 = (C(ID,N1)*H1+C(ID,N2)*H2+C(ID,N3)*H3+C(ID,N4)*(H1+H2)/2.
     1 +C(ID,N5)*(H2+H3)/2.+C(ID,N6)*(H1+H3)/2.)/(6.*HAVG)
      IF (CAVG1.LE.EPSI) GO TO 305
      CAVG5 = (C(ID+4,N1)*H1+C(ID+4,N2)*H2+C(ID+4,N3)*H3+C(ID+4,N4)*
     1 (H1+H2)/2.+C(ID+4,N5)*(H2+H3)/2.+C(ID+4,N6)*(H1+H3)/2.)/
     2 (6.*HAVG)
      RSAV1(M) = -SD(ID,M)*CAVG5/CAVG1
      GO TO 305
C
C     RESUSPENSION
C
150   SR(ID,M)=S2
C
C     TO COMPUTE A NUMBER OF LAYERS SCOURED IN ORDER TO RESUSPEND THE
C     APPROPRIATE AMOUNT OF SAND
C     ASSUME SAND IS MOST DIFFICULT TO SCOUR
      SR(ID,M)=SR(ID,M)*T
      ILAYR(M,ID)=0
      NB = NBED(M)
      IF (SR(ID,M).GT.XNT(M,ID)) GO TO 200
      RSAV1(M)=SR(ID,M)*GBD(M,NBED(M))
      XNT(M,ID)=XNT(M,ID)-SR(ID,M)
      GO TO 290
  200 ILAYR(M,ID)=ILAYR(M,ID)+1
      IF (ILAYR(M,ID).EQ.NBED(M)) GO TO 280
      RSAV1(M)=RSAV1(M)+XNT(M,ID)*GBD(M,NB)
      SR(ID,M)=SR(ID,M)-XNT(M,ID)
      RS=RS+XNT(M,ID)
      NB=NBED(M)-ILAYR(M,ID)
      TOTAL=GBA(M,NB)/RHOSED(1)+GBB(M,NB)/RHOSED(2)+GBC(M,NB)/RHOSED(3)
      XND=(1.-POR)/TOTAL
      XNT(M,ID)=BDIV(M)*GBA(M,NB)*XND
      IF (ILAYR(M,ID).LT.(NBED(M)-1)) GO TO 270
      ILAYR(M,ID)=ILAYR(M,ID)+1
      GO TO 280
270   IF (SR(ID,M).GE.XNT(M,ID)) GO TO 200
280   RSAV1(M)=RSAV1(M)+AMIN1(SR(ID,M),XNT(M,ID))*GBD(M,NB)
      RS=RS+AMIN1(SR(ID,M),XNT(M,ID))
      XNT(M,ID)=XNT(M,ID)-AMIN1(SR(ID,M),XNT(M,ID))
      SR(ID,M)=RS
290   SR(ID,M)=SR(ID,M)/T
      RSAV1(M)=RSAV1(M)/T
  305 CONTINUE
      RETURN
 6000 FORMAT(1X,"****WARNING***",/,
     $1X,"THE COMPUTED DEPOSITION RATE FOR ELEMENT ",I3," IS ",1PE12.4,
     $" KGF/DAY-M**2.")
 6001 FORMAT(1X,"THE THEORETICAL CAPACITY DIFFERENCE IS ",1PE12.4," KGF/
     $M**2",/,
     $1X,"THEREFORE THE TIME STEP SHOULD BE REDUCED TO A MAXIMUM OF ",
```

```
     $E12.4," DAY.")
      END
      SUBROUTINE SEDIME (M,AREA,ID)
C
C     THIS ROUTINE CALCULATES COEFFICIENTS OF DECAY AND
C     SOURCE TERMS IN THE SEDIMENT TRANSPORT CONVECTION-DIFFUSION
C     EQUATION
C
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
      COMMON /BLK9/DX(100),DY(100),EX(100),EY(100),ALFA(100),BETA(100)
     9             ,HS(100)
      COMMON /BLK10/PEL(6,6),SEL(6,6),REL(6)
      COMMON /BLK11/D50(3,100),DD50(100),SR(3,100),SD(3,100)
      COMMON /BLK17/QLATE(240),QSARA(7,100),QPNT(7,240),CRATE(3,100),
     1  CD(4,240)
      LEVEL   2,QLATE,QSARA,QPNT,CRATE,CD
C
C
      N1=NOD(M,1)
      N2=NOD(M,2)
      N3=NOD(M,3)
      N4=NOD(M,4)
      N5=NOD(M,5)
      N6=NOD(M,6)
      H1        = H(N1)
      H2        = H(N2)
      H3        = H(N3)
C
C***     DECAY TERM       ***
      ALFA(M)  = 0.0
C**   ALFA(M)  = QLATE(M)   IF QLATE(M) IS CONSTANT WITHIN A ELEMENT
C
C***     SOURCE OR SINK TERM       ***
C     AREA SOCECE QSARA(ID,M)
      A         = AREA*QSARA(ID,M)/60.
      B         = AREA*(SR(ID,M)-SD(ID,M))/3.
C
      REL(1)   = A*(2.*H1-H2-H3)
      REL(2)   = -A*(H1-2.*H2+H3)
      REL(3)   = -A*(H1+H2-2.*H3)
      REL(4)   = B+A*(8.*H1+8.*H2+4.*H3)
      REL(5)   = B+A*(4.*H1+8.*H2+8.*H3)
      REL(6)   = B+A*(8.*H1+4.*H2+8.*H3)
C
C     POINT SOURCE QPNT(ID,NDS)
      B         = 1.0
C
      REL(1)   = REL(1)               +B*((30.*H1+6.*H2+6.*H3)*QPNT(ID,N1)
     1-(4.*H1+4.*H2-H3)*QPNT(ID,N2)     -(4.*H1-H2+4.*H3)*QPNT(ID,N3)
     2+(12.*H1-8.*H2-4.*H3)*QPNT(ID,N4)-(4.*H1+12.*H2+12.*H3)*
     3QPNT(ID,N5)+(12.*H1-4.*H2-8.*H3)*QPNT(ID,N6))
      REL(2)   = REL(2)               +B*((4.*H1+4.*H2-H3)*QPNT(ID,N1)*(-1)
     1+(6.*H1+30.*H2+6.*H3)*QPNT(ID,N2)+(H1-4.*H2-4.*H3)*QPNT(ID,N3)
     2-(8.*H1-12.*H2+4.*H3)*QPNT(ID,N4)-(4.*H1-12.*H2+8.*H3)*QPNT(ID,N5)
     3-(12.*H1+4.*H2+12.*H3)*QPNT(ID,N6))
      REL(3)   = REL(3)              +B*((4.*H1-H2+4.*H3)*QPNT(ID,N1)*(-1)
     1+(H1-4.*H2-4.*H3)*QPNT(ID,N2)     +(6.*H1+6.*H2+30.*H3)*QPNT(ID,N3)
     2-(12.*H1+12.*H2+4.*H3)*QPNT(ID,N4)-(4.*H1+8.*H2-12.*H3)*
     3QPNT(ID,N5)-(8.*H1+4.*H2-12.*H3)*QPNT(ID,N6))
      REL(4)   = REL(4)               +B*((12.*H1-8.*H2-4.*H3)*QPNT(ID,N1)
     1-(8.*H1-12.*H2+4.*H3)*QPNT(ID,N2)-(12.*H1+12.*H2+4.*H3)*
```

```
     2QPNT(ID,N3)+(96.*H1+96.*H2+32.*H3)*QPNT(ID,N4)+(32.*H1+48.*H2+32.*
     3H3)*QPNT(ID,N5)+(48.*H1+32.*H2+32.*H3)*QPNT(ID,N6))
      REL(5)    = REL(5)    +B*((4.*H1+12.*H2+12.*H3)*QPNT(ID,N1)*(-1)
     1-(4.*H1-12.*H2+8.*H3)*QPNT(ID,N2)-(4.*H1+8.*H2-12.*H3)*QPNT(ID,N3)
     2+(32.*H1+48.*H2+32.*H3)*QPNT(ID,N4)+(32.*H1+96.*H2+96.*H3)*
     3QPNT(ID,N5)+(32.*H1+32.*H2+48.*H3)*QPNT(ID,N6))
      REL(6)    = REL(6)             +B*((12.*H1-4.*H2-8.*H3)*QPNT(ID,N1)
     1-(12.*H1+4.*H2+12.*H3)*QPNT(ID,N2)-(8.*H1+4.*H2-12.*H3)*
     2QPNT(ID,N3)+(48.*H1+32.*H2+32.*H3)*QPNT(ID,N4)+(32.*H1+32.*H2+48.
     3*H3)*QPNT(ID,N5)+(96.*H1+32.*H2+96.*H3)*QPNT(ID,N6))
C
      RETURN
      END
      SUBROUTINE SHEAR (M)
C
C     THIS ROUTINE CALCULATES BED SHEAR STRESS AND SHEAR VELOCITY FOR
C     A SEDIMENT LADEN FLOW
C     REF. HYDRAULICS OF SEDIMENT TRANSPORT BY W.H.GRAF, EQ.8.49
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
      COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)
      COMMON /BLK12/ACOF(6),U(6),V(6),D(2),AKJ(9,100),ALMBDA,RHOWAT,
     1              AKP(3)
C     USTAR.....SHEAR VELOCITY
C     STRESS(J).BED SHEAR STRESS
C     D50(J,M)..MEDIAN SIZE DIAMETER OF SEDIMENT  J=1,3    M=1,NE
C     BD50(J)...TOTAL AVERAGE OF BED SEDIMENT DIAMETER
C     RHOWAT    SPECIFIC WT. OF WATER IN KG(FORCE)/M**3  (1000 KG(F)/M**
C     DENSWT....WATER DENSITY IN KG(FORCE)-DAY**2/M**4
C     AKAPPA....KARMAN CONSTANT
C
      AKAPPA=0.4
      N1=NOD(M,1)
      N2=NOD(M,2)
      N3=NOD(M,3)
      TV1=SQRT(VX(N1)**2+VY(N1)**2)
      TV2=SQRT(VX(N2)**2+VY(N2)**2)
      TV3=SQRT(VX(N3)**2+VY(N3)**2)
      USTAR1= TV1  /(17.66+(ALOG10(H(N1)/(96.5*BD50(M))))*2.3/AKAPPA)
      USTAR2= TV2  /(17.66+(ALOG10(H(N2)/(96.5*BD50(M))))*2.3/AKAPPA)
      USTAR3= TV3  /(17.66+(ALOG10(H(N3)/(96.5*BD50(M))))*2.3/AKAPPA)
      DENSWT=RHOWAT/(9.8*(3600.*24.)**2)
      STRESS(1)=DENSWT*USTAR1**2
      STRESS(2)=DENSWT*USTAR2**2
      STRESS(3)=DENSWT*USTAR3**2
      RETURN
      END
      SUBROUTINE SILT (M,ID,T)
C
C     THIS SUBROUTINE COMPUTES THE AMOUNT OF RESUSPENSION,SR,OR
C     DIPOSITION,SD,OF SILT
C     FOR SYMBOLS, SEE SUBROUTINE CLAY
      COMMON /BLK1/CT(240),C(9,240)
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
      COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)
      COMMON /BLK13/WS(3,100),CRSTRS(3,100),CDSTRS(3,100),ERODA(3,100)
      COMMON /BLK14/GBA(100,10),GBB(100,10),GBC(100,10),GBD(100,10),
     1         GBE(100,10),GBF(100,10),GBG(100,10),POR
      COMMON /BLK15/ILAYR(100,3),XYSO(100),BDIV(100),NBED(100),BED(100),
     1           RHOSED(3),XNT(100,3),RSAV1(100),RSAV2(100),RSAV3(100)
      DIMENSION S1(3)
```

```
      DATA EPSI/1.0E-10/
      SD(ID,M)=0.0
      SR(ID,M)=0.0
      IF (ID.EQ.2)  RSAV2(M)=0.
      IF (ID.EQ.3) RSAV3(M)=0.0
      RS=0.0
      ILAYR(M,ID)=0
      TOTAL = GBA(M,NBED(M))/RHOSED(1)+GBB(M,NBED(M))/RHOSED(2)+
     $ GBC(M,NBED(M))/RHOSED(3)
      XDTOP=(1.-POR)/TOTAL
      IF(ID.EQ.2)  XNT(M,ID)=XYS0(M)*GBB(M,NBED(M))*XDTOP
      IF(ID.EQ.3)  XNT(M,ID)=XYS0(M)*GBC(M,NBED(M))*XDTOP
      DO 999 II=1,3
      S1(II)=0.0
      IF (STRESS(II).LE.CRSTRS(ID,M).AND.STRESS(II).GE.CDSTRS(ID,M)) GO
     1TO 999
      IF (STRESS(II).GT.CRSTRS(ID,M)) GO TO 100
C     DEPOSITION
      S1(II)=WS(ID,M)*C(ID,NOD(M,II))*(1.0-(STRESS(II)/CDSTRS(ID,M))
     1)/H(NOD(M,II))*(-1)
      GO TO 999
C
C      RESUSPENSION
C
100   S1(II)=ERODA(ID,M)*(STRESS(II)/CRSTRS(ID,M)-1.0)
999   CONTINUE
      S2=(S1(1)+S1(2)+S1(3))/3.0
      IF (S2) 300,305,302
 300  CONTINUE
C
C
C  SEDIMENT DEPOSITION
C
       SD(ID,M)=(-1)*S2
      N1        = NOD(M,1)
      N2        = NOD(M,2)
      N3        = NOD(M,3)
      N4        = NOD(M,4)
      N5        = NOD(M,5)
      N6        = NOD(M,6)
      H1        = H(N1)
      H2        = H(N2)
      H3        = H(N3)
      HAVG      = (H1+H2+H3)/3.0
      CAVG2     = (C(ID,N1)*H1+C(ID,N2)*H2+C(ID,N3)*H3+C(ID,N4)*(H1+H2)/
     1            2.0+C(ID,N5)*(H2+H3)/2.0+C(ID,N6)*(H1+H3)/2.0)/(6.0*
     2            HAVG)
       IF(CAVG2.LE.EPSI) GO TO 305
       CAVG6    =  (C(ID+4,N1)*H1+C(ID+4,N2)*H2+C(ID+4,N3)*H3
     1             +C(ID+4,N4)*(H1+H2)/2.0+C(ID+4,N5)*(H2+H3)/2.0
     2             +C(ID+4,N6)*(H1+H3)/2.0)/(6.0*HAVG)
       IF (ID.EQ.2) RSAV2(M) =-SD(ID,M)*CAVG6/CAVG2
       IF (ID.EQ.3) RSAV3(M) =-SD(ID,M)*CAVG6/CAVG2
       ILAYR(M,ID)=-1
       GO TO 305
 302   CONTINUE
C
C  SEDIMENT EROSION
C
      SR(ID,M)=S2
C
C     TO COMPUTE THE AVAILABILITY OF SILT IN BED LAYERS
```

```
C     NUMBER OF LAYERS SCOURED IS DETERMINED BY OTHER SEDIMENT(CLAY
C     OR SAND2)
C
      NB=NBED(M)
      SR(ID,M)=SR(ID,M)*T
      IF (SR(ID,M).GT.XNT(M,ID).AND.ILAYR(M,1).GT.0) GO TO 200
      SR(ID,M)=AMIN1(SR(ID,M),XNT(M,ID))
      XNT(M,ID)=XNT(M,ID)-SR(ID,M)
      IF(ID.EQ.2) RSAV2(M)=SR(ID,M)*GBE(M,NBED(M))
      IF(ID.EQ.3)  RSAV3(M)=SR(ID,M)*GBF(M,NBED(M))
      GO TO 290
  200 ILAYR(M,ID)=ILAYR(M,ID)+1
      IF(ID.EQ.2) RSAV2(M)=RSAV2(M)+XNT(M,ID)*GBE(M,NB)
      IF(ID.EQ.3) RSAV3(M)=RSAV3(M)+XNT(M,ID)*GBF(M,NB)
      SR(ID,M)=SR(ID,M)-XNT(M,ID)
      RS=RS+XNT(M,ID)
      NB=NBED(M)-ILAYR(M,ID)
      IF (NB.LE.0) GO TO 285
      TOTAL=GBA(M,NB)/RHOSED(1)+GBB(M,NB)/RHOSED(2)+GBC(M,NB)/RHOSED(3)
      XND=(1.-POR)/TOTAL
      IF(ID.EQ.2) XNT(M,ID)=BDIV(M)*GBB(M,NB)*XND
      IF(ID.EQ.3)  XNT(M,ID)=BDIV(M)*GBC(M,NB)*XND
      IF(ILAYR(M,ID) .GE. ILAYR(M,1)) GO TO 280
      IF (SR(ID,M).GE.XNT(M,ID)) GO TO 200
280   RS=RS+AMIN1(SR(ID,M),XNT(M,ID))
      IF(ID.EQ.2) RSAV2(M)=RSAV2(M)+AMIN1(SR(ID,M),XNT(M,ID))*GBE(M,NB)
      IF(ID.EQ.3) RSAV3(M)=RSAV3(M)+AMIN1(SR(ID,M),XNT(M,ID))*GBF(M,NB)
  285 XNT(M,ID)=XNT(M,ID)-AMIN1(SR(ID,M),XNT(M,ID))
      SR(ID,M)=RS
290   SR(ID,M)=SR(ID,M)/T
      IF(ID.EQ.2) RSAV2(M)=RSAV2(M)/T
      IF(ID.EQ.3) RSAV3(M)=RSAV3(M)/T
  305 CONTINUE
      RETURN
      END
      SUBROUTINE SMATRX (M,AREA,INO)
C
C     THIS SUBROUTINE CONSTRUCTS CONVECTION, DIFFUSION, AND DECAY
C     TERMS OF THE MATRIX (SEL) FOR THE M-TH TRIANGLAR ELEMENT
C
C**   THIS IS THE 3-NODE VERSION OF SMATRX.
C**   WITH LINEAR INTERPOLATING POLYNOMIALS.
C
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)
      COMMON /BLK7/NOD(240,6),X(240),Y(240)
      COMMON /BLK9/DX(100),DY(100),EX(100),EY(100),ALFA(100),BETA(100)
     9            ,HS(100)
      COMMON /BLK10/PEL(6,6),SEL(6,6),REL(6)
      COMMON /BLK12/ACOF(6),U(6),V(6),D(2),AKJ(9,100),ALMBDA,RHOWAT,
     1             AKP(3)
C
      DIMENSION SSEL(6,6)
C
      A1=ACOF(1)
      A2=ACOF(2)
      A3=ACOF(3)
      B1=ACOF(4)
      B2=ACOF(5)
      B3=ACOF(6)
C
      N1=NOD(M,1)
      N2=NOD(M,2)
```

```
      N3=NOD(M,3)
C
      D1=DX(M)/(60.0*AREA)
      D2=DY(M)/(60.0*AREA)
      U1=VX(N1)
      U2=VX(N2)
      U3=VX(N3)
      V1=VY(N1)
      V2=VY(N2)
      V3=VY(N3)
      RATE=ALFA(M)
C
      H1        = H(N1)
      H2        = H(N2)
      H3        = H(N3)
C**      DIFFUSION TERMS
      SEL(1,1) = (D1*B1*B1+D2*A1*A1)*(9.*H1+3.*H2+3.*H3)
      SEL(1,2) = -(D1*B1*B2+D2*A1*A2)*(2.*H1+2.*H2+H3)
      SEL(1,3) = -(D1*B1*B3+D2*A1*A3)*(2.*H1+H2+2.*H3)
      SEL(1,4) = (D1*B1*B1+D2*A1*A1)*(3.*H1-2.*H2-H3)
     1           +(D1*B1*B2+D2*A1*A2)*(14.*H1+3.*H2+3.*H3)
      SEL(1,5) = (D1*B1*B2+D2*A1*A2)*(3.*H1-H2-2.*H3)
     1           +(D1*B1*B3+D2*A1*A3)*(3.*H1-2.*H2-H3)
      SEL(1,6) = (D1*B1*B1+D2*A1*A1)*(3.*H1-H2-2.*H3)
     1           +(D1*B1*B3+D2*A1*A3)*(14.*H1+3.*H2+3.*H3)
      SEL(2,2) = (D1*B2*B2+D2*A2*A2)*(3.*H1+9.*H2+3.*H3)
      SEL(2,3) = -(D1*B2*B3+D2*A2*A3)*(H1+2.*H2+2.*H3)
      SEL(2,4) = (D1*B1*B2+D1*A1*A2)*(3.*H1+14.*H2+3.*H3)
     1           -(D1*B2*B2+D2*A2*A2)*(2.*H1-3.*H2+H3)
      SEL(2,5) = (D1*B2*B3+D2*A2*A3)*(3.*H1+14.*H2+3.*H3)
     1           -(D1*B2*B2+D2*A2*A2)*(H1-3.*H2+2.*H3)
      SEL(2,6) = -(D1*B1*B2+D2*A1*A2)*(H1-3.*H2+2.*H3)
     1           -(D1*B2*B3+D2*A2*A3)*(2.*H1-3.*H2+H3)
      SEL(3,3) = (D1*B3*B3+D2*A3*A3)*(3.*H1+3.*H2+9.*H3)
      SEL(3,4) = -(D1*B1*B3+D2*A1*A3)*(H1+2.*H2-3.*H3)
     1           -(D1*B2*B3+D2*A2*A3)*(2.*H1+H2-3.*H3)
      SEL(3,5) = (D1*B2*B3+D2*A2*A3)*(3.*H1+3.*H2+14.*H3)
     1           -(D1*B3*B3+D2*A3*A3)*(H1+2.*H2-3.*H3)
      SEL(3,6) = (D1*B1*B3+D2*A1*A3)*(3.*H1+3.*H2+14.*H3)
     1           -(D1*B3*B3+D2*A3*A3)*(2.*H1+H2-3.*H3)
      SEL(4,4) = (D1*B1*B1+D2*A1*A1)*(8.*H1+24.*H2+8.*H3)
     1           +(D1*B1*B2+D2*A1*A2)*(16.*H1+16.*H2+8.*H3)
     2           +(D1*B2*B2+D2*A2*A2)*(24.*H1+8.*H2+8.*H3)
      SEL(4,5) =  (D1*B1*B2+D2*A1*A2)*(4.*H1+8.*H2+8.*H3)
     1           +(D1*B1*B3+D2*A1*A3)*(8.*H1+24.*H2+8.*H3)
     2           +(D1*B2*B2+D2*A2*A2)*(8.*H1+4.*H2+8.*H3)
     3           +(D1*B2*B3+D2*A2*A3)*(8.*H1+8.*H2+4.*H3)
      SEL(4,6) =  (D1*B1*B1+D2*A1*A1)*(4.*H1+8.*H2+8.*H3)
     1           +(D1*B1*B2+D2*A1*A2)*(8.*H1+4.*H2+8.*H3)
     2           +(D1*B1*B3+D2*A1*A3)*(8.*H1+8.*H2+4.*H3)
     3           +(D1*B2*B3+D2*A2*A3)*(24.*H1+8.*H2+8.*H3)
      SEL(5,5) =  (D1*B2*B2+D2*A2*A2)*(8.*H1+8.*H2+24.*H3)
     1           +(D1*B2*B3+D2*A2*A3)*(8.*H1+16.*H2+16.*H3)
     2           +(D1*B3*B3+D2*A3*A3)*(8.*H1+24.*H2+8.*H3)
      SEL(5,6) =  (D1*B1*B2+D2*A1*A2)*(8.*H1+8.*H2+24.*H3)
     1           +(D1*B1*B3+D2*A1*A3)*(4.*H1+8.*H2+8.*H3)
     2           +(D1*B2*B3+D2*A2*A3)*(8.*H1+4.*H2+8.*H3)
     3           +(D1*B3*B3+D2*A3*A3)*(8.*H1+8.*H2+4.*H3)
      SEL(6,6) =  (D1*B1*B1+D2*A1*A1)*(8.*H1+8.*H2+24.*H3)
     1           +(D1*B1*B3+D2*A1*A3)*(16.*H1+8.*H2+16.*H3)
     2           +(D1*B3*B3+D2*A3*A3)*(24.*H1+8.*H2+8.*H3)
C
```

```
      DO 100 I=1,5
      II       = I+1
      DO 100 J=II,6
      SEL(J,I) = SEL(I,J)
100   CONTINUE
C
C
C
C**       ADVECTION TERMS
C
      BU11       =(B1*U1+A1*V1)/2520.
      BU12       =(B1*U2+A1*V2)/2520.
      BU13       =(B1*U3+A1*V3)/2520.
      BU21       =(B2*U1+A2*V1)/2520.
      BU22       =(B2*U2+A2*V2)/2520.
      BU23       =(B2*U3+A2*V3)/2520.
      BU31       =(B3*U1+A3*V1)/2520.
      BU32       =(B3*U2+A3*V2)/2520.
      BU33       =(B3*U3+A3*V3)/2520.
C
      SSEL(1,1) =                        +BU11*(102.*H1+12.*H2+12.*H3)
     1   +BU12*(12.*H1+6.*H2+3.*H3)      +BU13*(12.*H1+6.*H2+3.*H3)
      SSEL(2,1) =                        -BU11*(22.*H1+8.*H2+5.*H3)
     1   -BU12*(8.*H1+30.*H2+4.*H3)      -BU13*(5.*H1+4.*H2-2.*H3)
      SSEL(3,1) =                        -BU11*(22.*H1+5.*H2+8.*H3)
     1   -BU12*(4.*H1-2.*H2+5.*H3)       -BU13*(8.*H1+4.*H2+30.*H3)
      SSEL(4,1) =                        +BU11*(108.*H1+40.*H2+20.*H3)
     1   +BU12*(40.*H1+12.*H2+4.*H3)     +BU13*(20.*H1+4.*H2+4.*H3)
      SSEL(5,1) =                        +BU11*(20.*H1+4.*H2+4.*H3)
     1   +BU12*(4.*H1-36.*H2-24.*H3)     +BU13*(4.*H1-36.*H2-24.*H3)
      SSEL(6,1) =                        +BU11*(108.*H1+20.*H2+40.*H3)
     1   +BU12*(20.*H1+4.*H2+4.*H3)      +BU13*(40.*H1+4.*H2+12.*H3)
      SSEL(1,2) =                        -BU21*(30.*H1+8.*H2+4.*H3)
     1   -BU22*(8.*H1+22.*H2+5.*H3)      -BU23*(4.*H1+5.*H2-2.*H3)
      SSEL(2,2) =                        +BU21*(6.*H1+12.*H2+3.*H3)
     1   +BU22*(12.*H1+102.*H2+12.*H3)   +BU23*(3.*H1+12.*H2+6.*H3)
      SSEL(3,2) =                        +BU21*(2.*H1-5.*H2-4.*H3)
     1   -BU22*(5.*H1+22.*H2+8.*H3)      -BU23*(4.*H1+8.*H2+30.*H3)
      SSEL(4,2) =                        +BU21*(12.*H1+40.*H2+4.*H3)
     1   +BU22*(40.*H1+108.*H2+20.*H3)   +BU23*(4.*H1+20.*H2+4.*H3)
      SSEL(5,2) =                        +BU21*(4.*H1+20.*H2+4.*H3)
     1   +BU22*(20.*H1+108.*H2+40.*H3)   +BU23*(4.*H1+40.*H2+12.*H3)
      SSEL(6,2) =                        -BU21*(36.*H1-4.*H2+24.*H3)
     1   +BU22*(4.*H1+20.*H2+4.*H3)      -BU23*(24.*H1-4.*H2+36.*H3)
      SSEL(1,3) =                        -BU31*(30.*H1+4.*H2+8.*H3)
     1   -BU32*(5.*H1-2.*H2+4.*H3)       -BU33*(8.*H1+5.*H2+22.*H3)
      SSEL(2,3) =                        +BU31*(2.*H1-4.*H2-5.*H3)
     1   -BU32*(4.*H1+30.*H2+8.*H3)      -BU33*(5.*H1+8.*H2+22.*H3)
      SSEL(3,3) =                        +BU31*(6.*H1+3.*H2+12.*H3)
     1   +BU32*(3.*H1+6.*H2+12.*H3)      +BU33*(12.*H1+12.*H2+102.*H3)
      SSEL(4,3) =                        -BU31*(36.*H1+24.*H2-4.*H3)
     1   -BU32*(24.*H1+36.*H2-4.*H3)     +BU33*(4.*H1+4.*H2+20.*H3)
      SSEL(5,3) =                        +BU31*(4.*H1+4.*H2+20.*H3)
     1   +BU32*(4.*H1+12.*H2+40.*H3)     +BU33*(20.*H1+40.*H2+108.*H3)
      SSEL(6,3) =                        +BU31*(12.*H1+4.*H2+40.*H3)
     1   +BU32*(4.*H1+4.*H2+20.*H3)      +BU33*(40.*H1+20.*H2+108.*H3)
      SSEL(1,4) =                        +BU11*(12.*H1-8.*H2-4.*H3)
     1   +BU21*(144.*H1+12.*H2+12.*H3)   -BU12*(8.*H1+36.*H2+12.*H3)
     2   +BU22*(12.*H1-8.*H2-4.*H3)      -BU13*(4.*H1+12.*H2+12.*H3)
     3   +BU23*(12.*H1-4.*H2-8.*H3)
      SSEL(2,4) =                        -BU11*(8.*H1-12.*H2+4.*H3)
     1   -BU21*(36.*H1+8.*H2+12.*H3)     +BU12*(12.*H1+144.*H2+12.*H3)
```

```
     2   -BU22*(8.*H1-12.*H2+4.*H3)      -BU13*(4.*H1-12.*H2+8.*H3)        SMATRX
     3   -BU23*(12.*H1+4.*H2+12.*H3)                                       SMATRX
      SSEL(3,4) =                        -BU11*(12.*H1+12.*H2+4.*H3)       SMATRX
     1   -BU21*(36.*H1+12.*H2+8.*H3)     -BU12*(12.*H1+36.*H2+8.*H3)       SMATRX
     2   -BU22*(12.*H1+12.*H2+4.*H3)     -BU13*(4.*H1+8.*H2-12.*H3)        SMATRX
     3   -BU23*(8.*H1+4.*H2-12.*H3)                                        SMATRX
      SSEL(4,4) =                        +BU11*(96.*H1+96.*H2+32.*H3)      SMATRX
     1   +BU21*(192.*H1+96.*H2+48.*H3)   +BU12*(96.*H1+192.*H2+48.*H3)     SMATRX
     2   +BU22*(96.*H1+96.*H2+32.*H3)    +BU13*(32.*H1+48.*H2+32.*H3)      SMATRX
     3   +BU23*(48.*H1+32.*H2+32.*H3)                                      SMATRX
      SSEL(5,4) =                        +BU11*(32.*H1+48.*H2+32.*H3)      SMATRX
     1   +BU21*(48.*H1+32.*H2+32.*H3)    +BU12*(48.*H1+192.*H2+96.*H3)     SMATRX
     2   +BU22*(32.*H1+48.*H2+32.*H3)    +BU13*(32.*H1+96.*H2+96.*H3)      SMATRX
     3   +BU23*(32.*H1+32.*H2+48.*H3)                                      SMATRX
      SSEL(6,4) =                        +BU11*(48.*H1+32.*H2+32.*H3)      SMATRX
     1   +BU21*(192.*H1+48.*H2+96.*H3)   +BU12*(32.*H1+48.*H2+32.*H3)      SMATRX
     2   +BU22*(48.*H1+32.*H2+32.*H3)    +BU13*(32.*H1+32.*H2+48.*H3)      SMATRX
     3   +BU23*(96.*H1+32.*H2+96.*H3)                                      SMATRX
      SSEL(1,5) =                        +BU21*(12.*H1-4.*H2-8.*H3)        SMATRX
     1   +BU31*(12.*H1-8.*H2-4.*H3)      -BU22*(4.*H1+12.*H2+12.*H3)       SMATRX
     2   -BU32*(8.*H1+36.*H2+12.*H3)     -BU23*(8.*H1+12.*H2+36.*H3)       SMATRX
     3   -BU33*(4.*H1+12.*H2+12.*H3)                                       SMATRX
      SSEL(2,5) =                        -BU21*(12.*H1+4.*H2+12.*H3)       SMATRX
     1   -BU31*(8.*H1-12.*H2+4.*H3)      -BU22*(4.*H1-12.*H2+8.*H3)        SMATRX
     2   +BU32*(12.*H1+144.*H2+12.*H3)   -BU23*(12.*H1+8.*H2+36.*H3)       SMATRX
     3   -BU33*(4.*H1-12.*H2+8.*H3)                                        SMATRX
      SSEL(3,5) =                        -BU21*(8.*H1+4.*H2-12.*H3)        SMATRX
     1   -BU31*(12.*H1+12.*H2+4.*H3)     -BU22*(4.*H1+8.*H2-12.*H3)        SMATRX
     2   -BU32*(12.*H1+36.*H2+8.*H3)     +BU23*(12.*H1+12.*H2+144.*H3)     SMATRX
     3   -BU33*(4.*H1+8.*H2-12.*H3)                                        SMATRX
      SSEL(4,5) =                        +BU21*(48.*H1+32.*H2+32.*H3)      SMATRX
     1   +BU31*(96.*H1+96.*H2+32.*H3)    +BU22*(32.*H1+48.*H2+32.*H3)      SMATRX
     2   +BU32*(96.*H1+192.*H2+48.*H3)   +BU23*(32.*H1+32.*H2+48.*H3)      SMATRX
     3   +BU33*(32.*H1+48.*H2+32.*H3)                                      SMATRX
      SSEL(5,5) =                        +BU21*(32.*H1+32.*H2+48.*H3)      SMATRX
     1   +BU31*(32.*H1+48.*H2+32.*H3)    +BU22*(32.*H1+96.*H2+96.*H3)      SMATRX
     2   +BU32*(48.*H1+192.*H2+96.*H3)   +BU23*(48.*H1+96.*H2+192.*H3)     SMATRX
     3   +BU33*(32.*H1+96.*H2+96.*H3)                                      SMATRX
      SSEL(6,5) =                        +BU21*(96.*H1+32.*H2+96.*H3)      SMATRX
     1   +BU31*(48.*H1+32.*H2+32.*H3)    +BU22*(32.*H1+32.*H2+48.*H3)      SMATRX
     2   +BU32*(32.*H1+48.*H2+32.*H3)    +BU23*(96.*H1+48.*H2+192.*H3)     SMATRX
     3   +BU33*(32.*H1+32.*H2+48.*H3)                                      SMATRX
      SSEL(1,6) =                        +BU11*(12.*H1-4.*H2-8.*H3)        SMATRX
     1   +BU31*(144.*H1+12.*H2+12.*H3)   -BU12*(12.*H1+12.*H2+4.*H3)       SMATRX
     2   -BU32*(4.*H1+8.*H2-12.*H3)      -BU13*(8.*H1+12.*H2+36.*H3)       SMATRX
     3   +BU33*(12.*H1-4.*H2-8.*H3)                                        SMATRX
      SSEL(2,6) =                        -BU11*(12.*H1+4.*H2+12.*H3)       SMATRX
     1   -BU31*(36.*H1+8.*H2+12.*H3)     -BU12*(4.*H1-12.*H2+8.*H3)        SMATRX
     2   -BU32*(8.*H1-12.*H2+4.*H3)      -BU13*(12.*H1+8.*H2+36.*H3)       SMATRX
     3   -BU33*(12.*H1+4.*H2+12.*H3)                                       SMATRX
      SSEL(3,6) =                        -BU11*(8.*H1+4.*H2-12.*H3)        SMATRX
     1   -BU31*(36.*H1+12.*H2+8.*H3)     -BU12*(4.*H1+8.*H2-12.*H3)        SMATRX
     2   -BU32*(12.*H1+12.*H2+4.*H3)     +BU13*(12.*H1+12.*H2+144.*H3)     SMATRX
     3   -BU33*(8.*H1+4.*H2-12.*H3)                                        SMATRX
      SSEL(4,6) =                        +BU11*(48.*H1+32.*H2+32.*H3)      SMATRX
     1   +BU31*(192.*H1+96.*H2+48.*H3)   +BU12*(32.*H1+48.*H2+32.*H3)      SMATRX
     2   +BU32*(96.*H1+96.*H2+32.*H3)    +BU13*(32.*H1+32.*H2+48.*H3)      SMATRX
     3   +BU33*(48.*H1+32.*H2+32.*H3)                                      SMATRX
      SSEL(5,6) =                        +BU11*(32.*H1+32.*H2+48.*H3)      SMATRX
     1   +BU31*(48.*H1+32.*H2+32.*H3)    +BU12*(32.*H1+96.*H2+96.*H3)      SMATRX
     2   +BU32*(32.*H1+48.*H2+32.*H3)    +BU13*(48.*H1+96.*H2+192.*H3)     SMATRX
     3   +BU33*(32.*H1+32.*H2+48.*H3)                                      SMATRX
```

```
      SSEL(6,6) =                         +BU11*(96.*H1+32.*H2+96.*H3)     SMATRX   215
     1   +BU31*(192.*H1+48.*H2+96.*H3)  +BU12*(32.*H1+32.*H2+48.*H3)     SMATRX   216
     2   +BU32*(48.*H1+32.*H2+32.*H3)   +BU13*(96.*H1+48.*H2+192.*H3)    SMATRX   217
     3   +BU33*(96.*H1+32.*H2+96.*H3)                                     SMATRX   218
C                                                                         SMATRX   219
C**      DECAY TERMS                                                      SMATRX   220
C                                                                         SMATRX   221
C     A          = AREA/1260.                                             SMATRX   222
C                                                                         SMATRX   223
      DO 130 J=1,6                                                        SMATRX   224
      DO 130 I=1,6                                                        SMATRX   225
      SEL(I,J) = SEL(I,J)+SSEL(I,J)+RATE*PEL(I,J)                        SMATRX   226
130   CONTINUE                                                            SMATRX   227
      DO 140 J=1,6                                                        FETFIX2    1
      DO 140 I=1,6                                                        FETFIX2    2
      IF (I.EQ.J) GO TO 140                                               FETFIX2    3
      IF (SSEL(I,J).LE.0) GO TO 140                                       FETFIX2    4
      SEL(I,J) = SEL(I,J) - SSEL(I,J)                                     FETFIX2    5
      SEL(J,I) = SEL(J,I) - SSEL(I,J)                                     FETFIX2    6
      SEL(I,I) = SEL(I,I) + SSEL(I,J)                                     FETFIX2    7
      SEL(J,J) = SEL(J,J) + SSEL(I,J)                                     FETFIX2    8
140   CONTINUE                                                            FETFIX2    9
      RETURN                                                              SMATRX   228
      END                                                                 SMATRX   229
      FUNCTION SOL(H,G,W,TOL)                                             SOL        2
C                                                                         SOL        3
C**   THIS FUNCTION SUBPROGRAN SOLVES THE TRANSCENDENTAL EQ. FOR K.       SOL        4
C**   K=W**2/(G*TANH(K*H))                                                SOL        5
C                                                                         SOL        6
      C=G/W**2                                                            SOL        7
C                                                                         SOL        8
C**   MAKE INITIAL ESTIMATE OF K USING K=W/CS                             SOL        9
C**   WHERE CS IS THE SHALLOW WATER WAVE CELERITY.                        SOL       10
C                                                                         SOL       11
      DUMMY=W/SQRT(G*H)                                                   SOL       12
    1 WN=DUMMY                                                            SOL       13
      DUMMY=1./(C*TANH(WN*H))                                             SOL       14
      DIFF=ABS(WN-DUMMY)                                                  SOL       15
      IF(DIFF .GT. TOL) GO TO 1                                           SOL       16
      SOL=DUMMY                                                           SOL       17
      RETURN                                                              SOL       18
      END                                                                 SOL       19
      SUBROUTINE SPECS(NE,NDS,NCOL,NTP,T,ID,IDN,INO,KK,                   SPECS      2
     1 NPRNT,RSTRT,STORE,KKK,R,T)                                         SPECS      3
C                                                                         SPECS      4
C     THIS ROUTINE READS, CHECKS AND PRINTS THE PROBLEM SPECIFICATION.    SPECS      5
C                                                                         SPECS      6
C     THE UNITS OF THE INPUT DATA...METER,KILOGRAM(FORCE),DAY,PICOCURIE   SPECS      7
C     THE UNIT OF SEDIMENT CONC. ...(KG/CUBIC METER)                      SPECS      8
C     THE UNIT OF DISSOLVED CHEMICAL....(KG OR PCI/CUBIC METER OF WATER)  SPECS      9
C     THE UNIT OF CHEMICAL ATTACHED TO SED..(KG OR PCI/CUBIC METER OF WA  SPECS     10
C     THE OUTPUT UNIT OF CHEMICAL ATTACHED TO SEDIMENT(KG OF PCI/KG OF    SPECS     11
C        SEDIMENT)                                                        SPECS     12
C                                                                         SPECS     13
C                                                                         SPECS     14
C     CHARACTER*5 NAME,JCHECK,ZTYP,FLAG                                   FETFIX1   16
C     CHARACTER*10 SCHECK                                                 FETFIX1   17
      REAL K                                                              SPECS     17
C                                                                         SPECS     18
      COMMON /BLK1/CT(240),C(9,240)                                       SPECS     19
      COMMON /BLK4/ R(240), RPAST(240,7), NODBET, BETA1, AREA1            SPECS     20
      LEVEL  2,R,RPAST,NODBET,BETA1,AREA1                                 FETFIX4   22
```

```
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)                    SPECS
      COMMON /BLK7/NOD(240,6),X(240),Y(240)                            SPECS
      COMMON /BLK8/KBC,LBC,MBC(7,120),NBC(7,120),DBC(7,120),BC(7,120), SPECS
     1 KODE(10)                                                        SPECS
      COMMON /BLK9/DX(100),DY(100),EX(100),EY(100),ALFA(100),BETA(100) SPECS
     9           ,HS(100)                                              SPECS
      COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)           SPECS
      COMMON /BLK12/ACOF(6),U(6),V(6),D(2),AKJ(9,100),ALMBDA,RHOMAT,   SPECS
     1           AKP(3)                                                SPECS
      COMMON /BLK13/WS(3,100),CRSTRS(3,100),CDSTRS(3,100),ERODA(3,100) SPECS
      COMMON /BLK14/GBA(100,10),GBB(100,10),GBC(100,10),GBD(100,10),   SPECS
     1           GBE(100,10),GBF(100,10),GBG(100,10),POR               SPECS
      COMMON /BLK15/ILAYR(100,3),XYS0(100),BDIV(100),NBED(100),BED(100), SPECS
     1           RHOSED(3),XNT(100,3),RSAV1(100),RSAV2(100),RSAV3(100) SPECS
      COMMON /BLK16/ STYP                                              SPECS
      COMMON /BLK17/QLATE(240),QSARA(7,100),QPNT(7,240),CRATE(3,100),  SPECS
     1  CD(4,240)                                                      SPECS
      LEVEL   2,QLATE,QSARA,QPNT,CRATE,CD                              FETFIX4
C                                                                      SPECS
C                                                                      SPECS
      COMMON/WAVE/ NODE(240),  NC(135), A(135,10), K(135,10)           SPECS
     $           , W(135,10), HB(135), ALPHAC(135), WAVE, NNN, DD      SPECS
     $           , VIS, DPTHS(135)                                     SPECS
      COMMON /ZTYPE/ ZTYP(135)                                         SPECS
      COMMON/OPTION/ CLAY, SAND                                        SPECS
C                                                                      SPECS
      LOGICAL NUICS, NUCOF, NUVEL, TVBCS, TVHYD, TVINP,RSTRT,STORE,    SPECS
     $       USSDI, NSSDI, UBSDI, NBSDI, UBGEO, NBGEO, UBCON, NBCON    SPECS
      LOGICAL CLAY, SAND, WAVE, WAVR, VELR                             SPECS
      DIMENSION ARAD(3),ALEFT(3),B2(3),B3(3),GBJ(3),CS(3)              SPECS
      DIMENSION NAME(32), FLAG(16)                                     SPECS
      DATA KU/410/,MU/410/,NU/43/                                      SPECS
C                                                                      SPECS
      IF(KKK.EQ.1) GO TO 100                                           SPECS
   10 CONTINUE                                                         SPECS
      WRITE(6,888)KK,ID,NBED(74),RAT,T,C(ID,188),GBG(74,NBED(74))      SPECS
888   FORMAT(2X,3I5,4E12.5)                                            SPECS
      IF(KKK.EQ.1) WRITE(6,2070) ID,INO                                SPECS
      IF(.NOT.TVINP .AND. KKK.GT.1) RETURN                             SPECS
      IF(.NOT.TVBCS .AND. KKK.GT.1) GO TO 20                           SPECS
      GO TO 250                                                        SPECS
   20 CONTINUE                                                         SPECS
      IF(.NOT.TVHYD .AND. KKK.GT.1) RETURN                             SPECS
      IF(ID .GT. 1) RETURN                                             SPECS
      GO TO 270                                                        SPECS
C                                                                      SPECS
C        CARD 1-2                                                      SPECS
C     NAME(J)...TITLE OR COMMENT CARDS(2)                              SPECS
100   READ (5,310) (NAME(J),J=1,32)                                    SPECS
      WRITE (6,320)                                                    SPECS
      WRITE (6,520)                                                    SPECS
      WRITE (6,330) (NAME(J),J=1,32)                                   SPECS
C                                                                      SPECS
C        CARD 3                                                        SPECS
C     ID........IDENTIFICATION OF SIMULATION SUBSTANCES                SPECS
C     ID=1,2,3..SEDIMENT                                               SPECS
C     ID=4......DISOLVED CHEMICAL                                      SPECS
C     ID=5,6,7..CHEMICAL ADSORBED BY SEDIMENT                          SPECS
C     ID=8......TOTAL AMOUNT OF SEDIMENT                               SPECS
C     ID=9......TOTAL AMOUNT OF CHEMICAL                               SPECS
C     INO.......FLAG FOR COEFFICIENT SUBROUTINES FOR TRANSPORT CODE (=0) SPECS
C                                                                      SPECS
```

```
      READ (5,325) ID,INO
      IF (ID.GT.1) GO TO 125
C
C        CARD 4-5
C
C     IDN.......TOTAL NUMBER OF SIMULATION SUBSTANCES
C     NE........TOTAL NUMBER OF TRIANGULAR ELEMENTS
C     NDS.......TOTAL NUMBER OF NODES
C     NTP.......TOTAL NUMBER OF TIME PLANES
C               (=0 FOR STEADY-STATE SOL.)
C     NPRNT.....PRINT FREQUENCY
C     T.........SIMULATION TIME STEP
C**   STYP......NUMERICAL SOLUTION TYPE
C**             =0. IMPLIES EXPLICIT SOLUTION
C**             =.5 IMPLIES CRANK-NICHOLSON SOLUTION(DEFAULT).
C**             =1. IMPLIES IMPLICIT SOLUTION
      READ(5,341) IDN,NE,NDS,NTP,NPRNT
      READ(5,349) T,STYP,SCHECK
      IF(SCHECK .EQ. "          ") STYP=.5
      WRITE (6,520)
      WRITE (6,550) IDN,NE,NDS,NTP,T,NPRNT
      WRITE(6,2020) STYP
      IF(STYP .EQ. 0.) WRITE(6,2030)
      IF(STYP .EQ. .5) WRITE(6,2040)
      IF(STYP .EQ. 1.) WRITE(6,2050)
      IF((STYP .LT. 0.) .OR. (STYP .GT. 1.)) WRITE(6,2060)
      IF((STYP .LT. 0.) .OR. (STYP .GT. 1.)) STOP
      IF(NTP .EQ. 0) NPRNT=1
      IF (NDS.LT.MU.AND.NE.LT.KU.AND.NDS.GT.6) GO TO 110
      WRITE (6,370)
      STOP
110   CONTINUE
120   CONTINUE
      WRITE (6,520)
C
C
C       CARD GROUP 6-7
C
C     NUQSA...........NUMBER OF AREA SOURCES
C     QSARA(I,J).....SOURCE STRENGTH(PCI/CUBIC METER OF DISCHARGE/DAY)
C       I............CONSTITUENT NUMBER
C       J............SOURCE LOCATION (ELEMENT NUMBER)
C     NUQSP...........NUMBER OF POINT SOURCES
C     QPNT(I,J)......SOURCE STRENGTH(PCI/CUBIC METER OF DISCHARGE/DAY)
C       I............CONSTITUENT NUMBER
C       J............SOURCE LOCATION (NODE NUMBER)
C
      DO 700 J=1,NE
      DO 700 I=1,IDN
      QSARA(I,J)=0.0
 700  CONTINUE
      DO 701 J=1,NDS
      DO 701 I=1,IDN
      QPNT(I,J)=0.0
 701  CONTINUE
C
C
      READ(5,509) NUQSA
      WRITE(6,705) NUQSA
      IF (NUQSA.EQ.0) GO TO 704
      WRITE(6,706)
      DO 702 II=1,NUQSA
```

```
      READ(5,435) I,J,QSARA(I,J)                                        SPECS     144
      WRITE(6,707)  I,J,QSARA(I,J)                                      SPECS     145
  702 CONTINUE                                                          SPECS     146
 704  CONTINUE                                                          FETFIX1    19
      READ(5,509) NUQSP                                                 SPECS     147
      WRITE(6,708)  NUQSP                                               SPECS     148
      IF (NUQSP.EQ.0) GO TO 710                                         FETFIX1    20
      WRITE(6,709)                                                      SPECS     150
      DO 703 II=1,NUQSP                                                 SPECS     151
      READ(5,435) I,J,QPNT(I,J)                                         SPECS     152
      WRITE(6,707)  I,J,QPNT(I,J)                                       SPECS     153
  703 CONTINUE                                                          SPECS     154
 710  CONTINUE                                                          FETFIX1    21
      WRITE(6,520)                                                      SPECS     155
      WRITE(6,390)                                                      SPECS     156
C                                                                       SPECS     157
C         CARD 8                                                        SPECS     158
C     I.........ELEMENT NUMBER                                          SPECS     159
C     NOD(N,K)..ELEMENT-NODE TABLE (CONNECTIVITY)                       SPECS     160
      DO 170 N=1,NE                                                     SPECS     161
      READ (5,560) I,NOD(I,1),NOD(I,4),NOD(I,2),NOD(I,5),NOD(I,3),NOD(I,SPECS     162
     16)                                                                SPECS     163
      WRITE (6,570) I,(NOD(I,L),L=1,6)                                  SPECS     164
      IF (I.LT.1.OR.I.GT.NE) WRITE (6,540)                              SPECS     165
      IF (I.LT.1.OR.I.GT.NE) STOP                                       SPECS     166
 170  CONTINUE                                                          SPECS     167
C                                                                       SPECS     168
C     COMPUTE THE BAND WIDTH (2*NCOL-1)                                 SPECS     169
C                                                                       SPECS     170
      NCOL=1                                                            SPECS     171
      DO 180 N=1,NE                                                     SPECS     172
      DO 180 I=1,6                                                      SPECS     173
      DO 180 J=1,6                                                      SPECS     174
      NN=NOD(N,I)-NOD(N,J)+1                                            SPECS     175
 180  IF (NCOL-NN.LT.0) NCOL=NN                                         SPECS     176
      WRITE (6,400) NCOL                                                SPECS     177
      IF (NCOL.LE.NU) GO TO 190                                         SPECS     178
      WRITE (6,410)                                                     SPECS     179
      STOP                                                              SPECS     180
 190  CONTINUE                                                          SPECS     181
      WRITE (6,520)                                                     SPECS     182
      WRITE(6,610)                                                      SPECS     183
C                                                                       SPECS     184
C         CARD 9                                                        SPECS     185
C     J.........CORNER NODE NUMBER                                      SPECS     186
C     X(J)......X-COORDINATE OF J-TH NODE                               SPECS     187
C     Y(J)......Y-COORDINATE OF J-TH NODE                               SPECS     188
C**   INPUT INFORMATION FOR CORNER NODES ONLY. FETRA INTERPOLATES       SPECS     189
C**   BETWEEN THE CORNER NODES TO FIND THE LOCATION OF MIDPOINTS.       SPECS     190
C**   A BLANK CARD MUST FOLLOW THE LAST NODE DATA CARD.                 SPECS     191
C                                                                       SPECS     192
      NCNDS = 0                                                         SPECS     193
  192 READ(5,500) J, XJ, YJ, JCHECK                                     SPECS     194
      IF(JCHECK .EQ. "     ") GO TO 200                                 SPECS     195
      X(J) = XJ                                                         SPECS     196
      Y(J) = YJ                                                         SPECS     197
      NCNDS = NCNDS + 1                                                 SPECS     198
      WRITE(6,611) J, X(J), Y(J)                                        SPECS     199
      GO TO 192                                                         SPECS     200
 200  CONTINUE                                                          SPECS     201
C**      CARD 10                                                        SPECS     202
C                                                                       SPECS     203
```

```
C**   BY SPECIFYING THE FOLLOWING VALUES THE OPTIONS DESCRIBED
C**   ARE ACTIVATED(SEPARATE THE OPTIONS WITH A COMMA). IF NO OPTIONS
C**   ARE DESIRED, INPUT A BLANK CARD.
C
C**   NUCOF,      FOR NONUNIFORM COEFICIENTS(DX,DY,ALFA,BETA,MS-SEE
C**               CARDS 15,16,AND 17).
C**   NUVEL,      FOR NONUNIFORM VELOCITY AND/OR DEPTH.
C**   TVBCS,      FOR TIME VARYING BOUNDARY CONDITIONS.
C**   TVHYD,      FOR TIME VARYING HYDRAULICS.
C**   TVINP,      FOR TIME VARYING INPUT DATA.
C**   USSDI,      FOR UNIFORM SUSPENDED SEDIMENT DIAMETER.
C**   NSSDI,      FOR NONUNIFORM SUSPENDED SEDIMENT DIAMETER.
C**   UBSDI,      FOR UNIFORM BED SEDIMENT DIAMETER.
C**   NBSDI,      FOR NONUNIFORM BED SEDIMENT DIAMETER.
C**   UBGEO,      FOR UNIFORM BED CONFIGURATION.
C**   NBGEO,      FOR NONUNIFORM BED CONFIGURATION.
C**   UBCON,      FOR UNIFORM BED CONCENTRATION.
C**   NBCON,      FOR NONUNIFORM BED CONCENTRATION.
C
  125 CONTINUE
      NUCOF=.FALSE.
      NUVEL=.FALSE.
      TVINP=.FALSE.
      TVBCS=.FALSE.
      TVHYD=.FALSE.
      USSDI=.FALSE.
      NSSDI=.FALSE.
      UBSDI=.FALSE.
      NBSDI=.FALSE.
      UBGEO=.FALSE.
      NBGEO=.FALSE.
      UBCON=.FALSE.
      NBCON=.FALSE.
      RSTRT=.FALSE.
      STORE=.FALSE.
      READ(5,1025) (FLAG(I),I=1,12)
      DO 25 I=1,12
      IF(FLAG(I) .EQ. "     ") GO TO 27
      IF(FLAG(I) .EQ. "NUCOF") NUCOF=.TRUE.
      IF(FLAG(I) .EQ. "NUVEL") NUVEL=.TRUE.
      IF(FLAG(I) .EQ. "TVINP") TVINP=.TRUE.
      IF(FLAG(I) .EQ. "TVBCS") TVBCS=.TRUE.
      IF(FLAG(I) .EQ. "TVHYD") TVHYD=.TRUE.
      IF(FLAG(I) .EQ. "USSDI") USSDI=.TRUE.
      IF(FLAG(I) .EQ. "NSSDI") NSSDI=.TRUE.
      IF(FLAG(I) .EQ. "UBSDI") UBSDI=.TRUE.
      IF(FLAG(I) .EQ. "NBSDI") NBSDI=.TRUE.
      IF(FLAG(I) .EQ. "UBGEO") UBGEO=.TRUE.
      IF(FLAG(I) .EQ. "NBGEO") NBGEO=.TRUE.
      IF(FLAG(I) .EQ. "UBCON") UBCON=.TRUE.
      IF(FLAG(I) .EQ. "NBCON") NBCON=.TRUE.
      IF(FLAG(I) .EQ. "RSTRT") RSTRT=.TRUE.
      IF(FLAG(I) .EQ. "STORE") STORE=.TRUE.
   25 CONTINUE
   27 CONTINUE
      WRITE(6,520)
      WRITE(6,6000)
      IF(NUCOF) WRITE(6,6015)
      IF(NUVEL) WRITE(6,6020)
      IF(TVBCS) WRITE(6,6025)
      IF(TVHYD) WRITE(6,6030)
      IF(TVINP) WRITE(6,6035)
```

```
      IF(USSDI) WRITE(6,6040)
      IF(NSSDI) WRITE(6,6045)
      IF(UBSDI) WRITE(6,6050)
      IF(NBSDI) WRITE(6,6055)
      IF(UBGEO) WRITE(6,6060)
      IF(NBGEO) WRITE(6,6065)
      IF(UBCON) WRITE(6,6070)
      IF(NBCON) WRITE(6,6075)
C
C**       CARD 10A
C
C**   BY SPECIFYING THE FOLLOWING VALUES THE OPTIONS DESCRIBED ARE
C**   ACTIVATED(SEPARATE THE OPTIONS WITH A COMMA), IF NO OPTIONS
C**   ARE DESIRED INPUT A BLANK CARD.
C
C**   CLAY,       REQUIRED IF INO(SEE CARD 3) IS GREATER THAN ZERO
C**               AND CLAY IS THE LIMITING BED LAYER.
C**   SAND,       REQUIRED IF INO(SEE CARD 3) IS GREATER THAN ZERO
C**               AND SAND IS THE LIMITING BED LAYER.
C**   WAVE,       REQUIRED IF THE CONTAMINANT AND/OR SEDIMENT TRANSPORT
C**               AT ANY OF THE NODES IS INFLUENCED BY A WAVE OR SURF
C**               ENVIRONMENT.
C**   WAVR,       REQUIRED IF WAVE, IS SPECIFIED AND THE WAVE AND SURF
C**               CHARACTERISTICS ARE TO BE READ FROM FILES CREATED BY
C**               OTHER PROGRAMS.
C**   VELR,       REQUIRED IF THE VELOCITY FIELD IS TO BE READ FROM FILES
C**               CREATED FROM OTHER PROGRAMS.
C
      CLAY=.FALSE.
      WAVR= .FALSE.
      SAND=.FALSE.
      WAVE= .FALSE.
      VELR = .FALSE.
      READ(5,1030) (FLAG(I),I=1,16)
      DO 30 I=1,10
      IF(FLAG(I) .EQ. "     ") GO TO 35
      IF(FLAG(I) .EQ. "CLAY,") CLAY=.TRUE.
      IF(FLAG(I) .EQ. "SAND,") SAND=.TRUE.
      IF(FLAG(I) .EQ. "WAVE,") WAVE=.TRUE.
      IF(FLAG(I) .EQ. "WAVR,") WAVR=.TRUE.
      IF(FLAG(I) .EQ. "VELR,") VELR=.TRUE.
30    CONTINUE
   35 CONTINUE
      IF (INO.LE.0) GO TO 107
C
C         CARD 10-12
C
C     D50(J,M)..SUSPENDED SEDIMENT DIAMETER SIZE   J=1,2,3
C     BD50(M)...BED SEDIMENT SIZE
C     CRSTRS(ID,M).CRITICAL SHEAR STRESS FOR SCOURING,ID=1,2,3
C     CDSTRS(ID,M).CRITICAL SHEAR STRESS FOR DEPOSITION ID=1,2,3
C     WS(ID,M).....FALL VELOCITY OF SEDIMENT ID=1,2,3 AT ELEMENT M
C     ERODA(ID,M)..ERODABILITY COEFFICIENT ID=1,2,3  AT ELEMENT M
C
      IF (ID.GT.3) GO TO 107
      IF(USSDI) GO TO 103
      IF(NSSDI) GO TO 108
      GO TO 104
108   READ (5,503) (D50(ID,M),WS(ID,M),CRSTRS(ID,M),CDSTRS(ID,M),
     1ERODA(ID,M),M=1,NE)
      WRITE(6,352)
      WRITE(6,353) (ID,M,D50(ID,M),WS(ID,M),CRSTRS(ID,M),CDSTRS(ID,M),
```

```
     1ERODA(ID,M),M=1,NE)                                               SPECS
      GO TO 104                                                         SPECS
103   READ (5,503) D50(ID,1),WS(ID,1),CRSTRS(ID,1),CDSTRS(ID,1),        SPECS
     1ERODA(ID,1)                                                       SPECS
      WRITE(6,352)                                                      SPECS
      WRITE(6,355) ID,D50(ID,1),WS(ID,1),CRSTRS(ID,1),CDSTRS(ID,1),     SPECS
     1ERODA(ID,1)                                                       SPECS
      DO 140 M=2,NE                                                     SPECS
      D50(ID,M)=D50(ID,1)                                               SPECS
      WS(ID,M)=WS(ID,1)                                                 SPECS
      CRSTRS(ID,M)=CRSTRS(ID,1)                                         SPECS
      CDSTRS(ID,M)=CDSTRS(ID,1)                                         SPECS
      ERODA(ID,M)=ERODA(ID,1)                                           SPECS
140   CONTINUE                                                          SPECS
      IF (ID.GT.1) GO TO 107                                            SPECS
  104 IF(UBSDI) GO TO 105                                               SPECS
      IF(NBSDI) GO TO 109                                               SPECS
      GO TO 107                                                         SPECS
109   READ (5,503) (BD50(M),M=1,NE)                                     SPECS
      GO TO 107                                                         SPECS
105   READ (5,505) BD50(1)                                              SPECS
      DO 106 I=1,NE                                                     SPECS
      BD50(I)=BD50(1)                                                   SPECS
106   CONTINUE                                                          SPECS
107   CONTINUE                                                          SPECS
C                                                                       SPECS
C        CARD 13-14                                                     SPECS
C                                                                       SPECS
C     UNICS,    REQUIRED FOR UNIFORM INITIAL CONDITIONS; THE CARD FOLLOW SPECS
C               THIS SHOULD INCLUDE C(I,1) WHERE I=1. THIS 2-CARD-SET IS SPECS
C               REPEATED FOR I=2,9                                      SPECS
C                                                                       SPECS
C               IF FLAG(1) DOES NOT EQUAL "UNICS," NONUNIFORM INITIAL CO SPECS
C               DITIONS ARE ASSUMED.  C(I,J) FOR J=1,NDS IS READ FOR THI SPECS
C               CASE.                                                   SPECS
C     C(ID,J)...INITIAL CONDITION AND SOL. ALREADY SOLVED AT PRESENT    SPECS
C               TIME STEP OR SOL. AT PREVIOUS TIME STEP                 SPECS
C                                                                       SPECS
      IF (ID.GT.1) GO TO 205                                            SPECS
      DO 50 I=1,9                                                       SPECS
      DO 50 J=1,NDS                                                     SPECS
   50 C(I,J)=0.0                                                        SPECS
      DO 150 I=ID,IDN                                                   SPECS
      READ(5,1025) FLAG(1)                                              SPECS
      WRITE(6,520)                                                      SPECS
      WRITE(6,6100) I                                                   SPECS
      IF(FLAG(1) .EQ. "UNICS") GO TO 130                                SPECS
      WRITE(6,6110)                                                     SPECS
      READ (5,510)  (C(I,J),J=1,NDS)                                    SPECS
      GO TO 150                                                         SPECS
130   READ (5,505) C(I,1)                                               SPECS
      WRITE(6,6120)                                                     SPECS
      DO 145 J=1,NDS                                                    SPECS
145   C(I,J)=C(I,1)                                                     SPECS
150   CONTINUE                                                          SPECS
205   WRITE (6,420)                                                     SPECS
      WRITE (6,490)                                                     SPECS
      NPRT=NDS/6+1                                                      SPECS
      DO 160 I=1,NPRT                                                   SPECS
      NST=(I-1)*6+1                                                     SPECS
      IF (NST.GT.NDS) GO TO 160                                         SPECS
      II=NST+5                                                          SPECS
```

```
      IF (II.GT.NDS) II=NDS                                             SPECS     390
      WRITE (6,530) (J,C(ID,J),J=NST,II)                                SPECS     391
160   CONTINUE                                                          SPECS     392
C                                                                       SPECS     393
      IF(ID .GT. 1) GO TO 10                                            SPECS     394
      IF(.NOT. NUCOF) GO TO 210                                         SPECS     395
C                                                                       SPECS     396
C        CARD 15-16-17                                                  SPECS     397
C                                                                       SPECS     398
C     DX(J).....DISPERSION COEFFICIENT - X COMPONENT                    SPECS     399
C     DY(J).....DISPERSION COEFFICIENT - Y COMPONENT                    SPECS     400
C     ALMBDA....DECAY RATE                                              SPECS     401
C     AKJ(I)....ADSORPTION OR DESORPTION RATE, I=1,9                    SPECS     402
C     AKP(I)....EQUAL 1 IF ADSORPTION OR DESORPTION OCCURS              SPECS     403
C               EQUAL 0 IF NO ADSORPTION OR DESORPTION OCCURS           SPECS     404
C     RHOSED(J).SPECIFIC WT. OF J-TH SEDIMENT                           SPECS     405
C     POR.......POROSITY                                                SPECS     406
C     RHOWAT....SPECIFIC WT. OF WATER(1000KG(FORCE)/M**3)               SPECS     407
C**   VIS.......KINEMATIC VISCOSITY OF WATER, M**2/SEC                  SPECS     408
C     ALFA (J)..DECAY TERM                                              SPECS     409
C     BETA(J)...SOURCE OR SINK TERM                                     SPECS     410
C     HS(J).....ELEMENT THICKNESS                                       SPECS     411
C                                                                       SPECS     412
      IF (INO.LE.0) GO TO 208                                           SPECS     413
      READ(5,351) ALMBDA,(AKP(I),I=1,3),BETA1,AREA1,NODBET              SPECS     414
      DO 206 I=1,9                                                      SPECS     415
      READ(5,336) (AKJ(I,J),J=1,NE)                                     SPECS     416
  206 CONTINUE                                                          SPECS     417
      READ(5,350) (RHOSED(I),I=1,3), RHOWAT, POR, VIS                   SPECS     418
      WRITE(6,383)                                                      SPECS     419
      WRITE(6,381) ALMBDA,AKP(1),AKP(2),AKP(3),BETA1,AREA1,NODBET       FETFIX1    22
      DO 207 I=1,9                                                      SPECS     421
      WRITE(6,385) I                                                    SPECS     422
      WRITE(6,386) (AKJ(I,J),J=1,NE)                                    SPECS     423
  207 CONTINUE                                                          SPECS     424
      WRITE(6,384)                                                      SPECS     425
      WRITE(6,381) (RHOSED(I),I=1,3), RHOWAT, POR, VIS                  SPECS     426
208   READ (5,350)(DX(J),DY(J),ALFA(J),BETA(J),HS(J),J=1,NE)            SPECS     427
      WRITE (6,380) (J,DX(J),DY(J),ALFA(J),BETA(J),HS(J),               SPECS     428
     1J=1,NE)                                                           SPECS     429
      WRITE (6,520)                                                     SPECS     430
      GO TO 240                                                         SPECS     431
C                                                                       SPECS     432
210   IF (INO.LE.0) GO TO 209                                           SPECS     433
      READ(5,351) ALMBDA,(AKP(I),I=1,3),BETA1,AREA1,NODBET              SPECS     434
      DO 212 I=1,9                                                      SPECS     435
      READ(5,336) AKJ(I,1)                                              SPECS     436
      DO 212 J=1,NE                                                     SPECS     437
      AKJ(I,J) = AKJ(I,1)                                               SPECS     438
  212 CONTINUE                                                          SPECS     439
      READ(5,350) (RHOSED(I),I=1,3), RHOWAT, POR, VIS                   SPECS     440
      WRITE(6,383)                                                      SPECS     441
      WRITE(6,381) ALMBDA,AKP(1),AKP(2),AKP(3),BETA1,AREA1,NODBET       FETFIX1    23
      DO 213 I=1,9                                                      SPECS     443
      WRITE(6,385) I                                                    SPECS     444
      WRITE(6,386) (AKJ(I,J),J=1,NE)                                    SPECS     445
  213 CONTINUE                                                          SPECS     446
      WRITE(6,384)                                                      SPECS     447
      WRITE(6,381) (RHOSED(I),I=1,3), RHOWAT, POR, VIS                  SPECS     448
209   READ (5,350) DX(1),DY(1),ALFA(1),BETA(1),HS(1)                    SPECS     449
      WRITE (6,580)                                                     SPECS     450
      WRITE (6,380) NE,DX(1),DY(1),ALFA(1),BETA(1),HS(1)                SPECS     451
```

```
      WRITE (6,520)
      DO 222 J=1,NE
      DX(J)=DX(1)
      DY(J)=DY(1)
      ALFA(J)=ALFA(1)
      BETA(J)=BETA(1)
      HS(J)=HS(1)
222   CONTINUE
240   CONTINUE
      IF(INO .LE. 0) GO TO 10
C
C**   CARD 18
C
C**   FOR ELEMENT M-
C**   NBED(M) =INITIAL NUMBER OF BED LAYERS.
C**   BDIV(M) =THICKNESS OF BED LAYERS(USED FOR ALL LAYERS
C**            EXCEPT TOP LAYER).
C**   XYSO(M) =THICKNESS OF TOP BED LAYER.
C**   BED(M)  =INITIAL TOTAL BED THICKNESS.
C
      IF(UBGEO) GO TO 122
      IF(NBGEO) GO TO 121
      GO TO 124
121   READ(5,335) (NBED(M),BDIV(M),XYSO(M),BED(M),M=1,NE)
      WRITE(6,532)
      WRITE(6,535)(M,NBED(M),BDIV(M),XYSO(M),BED(M),BD50(M),M=1,NE)
      GO TO 126
122   READ(5,335) NBED(1),BDIV(1),XYSO(1),BED(1)
      WRITE(6,531)
      WRITE(6,537)NBED(1),BDIV(1),XYSO(1),BED(1),BD50(1)
      DO 131 M=1,NE
      NBED(M)=NBED(1)
      BDIV(M)=BDIV(1)
      XYSO(M)=XYSO(1)
      BED(M)=BED(1)
131   CONTINUE
  126 IF(UBCON) GO TO 129
      IF(NBCON) GO TO 128
      GO TO 124
C
C**   CARD 19
C
C**   ***NOTE=BED LAYERS ARE NUMBERED FROM DEEPEST TO SHALLOWEST***
C
C**   FOR ELEMENT M, BED LAYER J-
C**   GBA(M,J)=WEIGHT FRACTION OF CLAY IF CLAY, IS THE
C**            LIMITING BED LAYER.
C**           =WEIGHT FRACTION OF SAND IF SAND IS THE
C**            THE LIMITING BED LAYER.
C**   GBB(M,J)=WEIGHT FRACTION OF SILT.
C**   GBC(M,J)=WEIGHT FRACTION OF SAND IF CLAY IS THE
C**            LIMITING BED LAYER.
C**           =WEIGHT FRACTION OF SILT IF SAND IS THE
C**            LIMITING BED LAYER.
C**   GBD(M,J)=CHEMICAL CONCENTRATION OF CONTAMINANT PER UNIT WEIGHT
C**            OF SEDIMENT(CLAY OR SAND DEPENDING ON THE SEDIMENT TYPE
C**            THAT GBA(M,J) APPLIES TO).
C**   GBE(M,J)=CHEMICAL CONCENTRATION OF CONTAMINANT PER UNIT
C**            WEIGHT OF SILT.
C**   GBF(M,J)=CHEMICAL CONCENTRATION OF CONTAMINANT PER UNIT WEIGHT
C**            OF SEDIMENT(CLAY OR SILT DEPENDING ON THE SEDIMENT TYPE
C**            THAT GBC(M,J) APPLIES TO).
```

```
C
C**   ***NOTE-APPROPRIATE UNITS FOR GBD, GBE, AND GBF ARE
C**            PCI/KG OR KG/KG.***
C
128   DO 127 M=1,NE
      NBED3=NBED(M)
      READ(5,336)  (GBA(M,J),GBB(M,J),GBC(M,J),GBD(M,J),GBE(M,J),GBF(M,J
     1),J=1,NBED3)
127   CONTINUE
      DO 195 M=1,NE
      NBED3=NBED(M)
      DO 195 J=1,NBED3
      GBG(M,J)=GBA(M,J)*GBD(M,J)+GBB(M,J)*GBE(M,J)+GBC(M,J)*GBF(M,J)
195   CONTINUE
      WRITE(6,533)
      DO 196 M=1,NE
      NBED3=NBED(M)
      WRITE(6,536)  (J,GBA(M,J),GBB(M,J),GBC(M,J),GBD(M,J),GBE(M,J),
     1GBF(M,J),GBG(M,J),J=1,NBED3)
196   CONTINUE
      GO TO 124
129   READ(5,336) GBA(1,1),GBB(1,1),GBC(1,1),GBD(1,1),GBE(1,1),GBF(1,1)
      GBG(1,1)=GBA(1,1)*GBD(1,1)+GBB(1,1)*GBE(1,1)+GBC(1,1)*GBF(1,1)
221   DO 123 M=1,NE
      NBEDM=NBED(1)
      DO 123 J=1,NBEDM
      GBA(M,J)=GBA(1,1)
      GBB(M,J)=GBB(1,1)
      GBC(M,J)=GBC(1,1)
      GBD(M,J)=GBD(1,1)
      GBE(M,J)=GBE(1,1)
      GBF(M,J)=GBF(1,1)
      GBG(M,J)=GBG(1,1)
123   CONTINUE
      WRITE(6,539)
      WRITE(6,538) GBA(1,1),GBB(1,1),GBC(1,1),GBD(1,1),GBE(1,1),GBF(1,1)
     1,GBG(1,1)
124   CONTINUE
      GO TO 10
C
C        CARD 20
C
C     READ THE BOUNDARY CONDITIONS
C
C     LBC.......NUMBER OF SPECIFIED BOUNDARY CONDITION NODES
C     KBC.......NUMBER OF DERIVATIVE BOUNDARY CONDITION NODES
C     NBC(ID,J)....BOUNDARY NODE NUMBER
C     BC(ID,J).....BOUNDARY NODE VALUE (SPECIFIED)
C     MBC(ID,J)....BOUNDARY NODE NUMBER
C     DBC(ID,J).....BOUNDARY NODE VALUE (DERIVATIVE)
C
250   CONTINUE
      READ(5,341)LBC,KBC
      JMAX=MAX0(KBC,LBC)
      IF(JMAX .LE. 0) GO TO 20
      NCHECK=1
      MCHECK=1
      IF(LBC .EQ. 0) NCHECK = 0
      IF(KBC .EQ. 0) MCHECK = 0
      DO 260 J=1,JMAX
      READ(5,600) NBC(ID,J),BC(ID,J),MBC(ID,J),DBC(ID,J)
C
```

```
C**    B.C. NODES MUST BE INPUT IN A NUMERICALLY INCREASING SEQUENCE         SPECS     576
C                                                                            SPECS     577
       IF(NBC(ID,J) .LT. NCHECK) WRITE(6,2000) NBC(ID,J)                     SPECS     578
       IF(MBC(ID,J) .LT. MCHECK) WRITE(6,2010) MBC(ID,J)                     SPECS     579
       NCHECK=NBC(ID,J)                                                      SPECS     580
       MCHECK=MBC(ID,J)                                                      SPECS     581
260    CONTINUE                                                              SPECS     582
       WRITE(6,460)                                                          SPECS     583
       WRITE (6,490)                                                         SPECS     584
       WRITE (6,470) (NBC(ID,J),BC(ID,J),J=1,LBC)                            SPECS     585
       IF(KBC .LE. 0) GO TO 20                                               SPECS     586
       WRITE (6,590)                                                         SPECS     587
       WRITE (6,470) (MBC(ID,J),DBC(ID,J),J=1,KBC)                           SPECS     588
  265 CONTINUE                                                               SPECS     589
C                                                                            SPECS     590
       GO TO 20                                                              SPECS     591
270    CONTINUE                                                              SPECS     592
       IF(ID.EQ.1.AND.KK.EQ.1.AND.RSTRT) READ(1) RAT,KK,C,NBED,              FETFIX1    24
      1 XYSO,BED,GBA,GBB,GBC,GBD,GBE,GBF,GBG                                 SPECS     594
       DO 275 J=1,NOS                                                        SPECS     595
       H(J) = 0.0                                                            SPECS     596
       VX(J) = 0.0                                                           SPECS     597
       VY(J) = 0.0                                                           SPECS     598
275    CONTINUE                                                              SPECS     599
       IF(.NOT. NUVEL) GO TO 280                                             SPECS     600
C                                                                            SPECS     601
C      READ THE FLOW FIELD                                                   SPECS     602
C                                                                            SPECS     603
C         CARD 21                                                            SPECS     604
C                                                                            SPECS     605
C      VX(J).....X-COMPONENT OF VELOCITY                                     SPECS     606
C      VY(J).....Y-COMPONENT OF VELOCITY                                     SPECS     607
C      H(J)......FLOW DEPTH                                                  SPECS     608
       IF (VELR) GO TO 295                                                   SPECS     609
       READ(5,430) (H(J),VX(J),VY(J),J=1,NOS)                                SPECS     610
       GO TO 300                                                             SPECS     611
  280 CONTINUE                                                               SPECS     612
       READ(5,430) H(1),VX(1),VY(1)                                          SPECS     613
       DO 290 J=1,NOS                                                        SPECS     614
       H(J)=H(1)                                                             SPECS     615
       VX(J)=VX(1)                                                           SPECS     616
       VY(J)=VY(1)                                                           SPECS     617
290    CONTINUE                                                              SPECS     618
       GO TO 300                                                             SPECS     619
295    IF(ID.EQ.1.AND.KKK.EQ.1) READ(5,336) VFREQ                            SPECS     620
       AT = RAT+T*(KKK-1)                                                    SPECS     621
       IF (ID.EQ.1 .AND. AMOD(AT,VFREQ) .LT. T) CALL RVEL(AT,RAT,            SPECS     622
      1 NCNDS,VFREQ)                                                         SPECS     623
  300 CONTINUE                                                               SPECS     624
       IF (MOD(KK,NPRNT).NE.0 .AND. KK.NE.NTP) GO TO 305                     SPECS     625
       WRITE(6,440)                                                          SPECS     626
       WRITE(6,480)                                                          SPECS     627
       NOSM2 = NOS - 2                                                       SPECS     628
       WRITE(6,450) (J,VX(J),VY(J),H(J),J+1,VX(J+1),VY(J+1),H(J+1),          SPECS     629
      1 J+2,VX(J+2),VY(J+2),H(J+2),J=1,NOSM2,3)                              SPECS     630
  305 CONTINUE                                                               SPECS     631
C                                                                            SPECS     632
C      IF (KK.EQ.2) WRITE (6,7005) KK,ID,(FLAG(IJK),IJK=1,6)                 SPECS     633
C      IF (KK.EQ.1.AND.ID.EQ.7) WRITE (6,7005) KK,ID,(FLAG(IJK),IJK=1,6)     SPECS     634
C7005 FORMAT (2I5,6(1X,A5))                                                  SPECS     635
       IF(.NOT. WAVE) GO TO 309                                              SPECS     636
       IF (WAVR) GO TO 312                                                   SPECS     637
```

```
C
C**      CARD 22,A,B,C
C
C**   THIS SEGMENT READS THE DATA REQUIRED FOR CALCULATION OF
C**   WAVE SEDIMENT TRANSPORT.  INPUT DATA AT THE CORNER NODES ONLY.
C
C**   NODNO    =THE NODE NUMBER(CORNER NODES ONLY).
C**   ZTYP     =WAVE ZONE TYPE
C**            =WAVE,  IF NODE IS BEYOND SURF ZONE
C**            =SURF,  IF NODE IS IN SURF ZONE
C**            =WAVC, IF NODE IS IN A WAVE ZONE AND THE WAVE
C**             CHARACTERISTICS ARE TO BE CALCULATED.
C**   NC       =NUMBER OF WAVE CHARACTERISTICS TO READ FOR
C**             THIS NODE(DEFAULT = 1).
C**   WV       =WIND VELOCITY, M/SEC.
C**   DM       =MEAN FETCH DEPTH, M.
C**   F        =EFFECTIVE FETCH LENGTH, M.
C**   A        =WAVE AMPLITUDE, M.
C**   K        =WAVE NUMBER=2*PI/WAVE LENGTH, 1./M.
C**   W        =WAVE FREQUENCY=2*PI/WAVE PERIOD, 1./SEC.
C**   HB       =WAVE HEIGHT AT BREAKING, M.
C**   ALPHAC   =ANGLE BETWEEN WAVE RAY AND THE GRADIENT OF THE
C**             BOTTOM BATHYMETRY, DEGREES.
C
      READ(5,1025) FLAG(1)
      WRITE(6,520)
      IF (FLAG(1) .EQ. "VWIND") GO TO 1315
      WRITE(6,6125)
      READ(5,6126)  WNDVEL
      WRITE(6,6127)  WNDVEL
      GO TO 6131
 1315 WRITE(6,6130)
 6131 WRITE(6,2075)
      DO 306 I=1,NDS
  306 NODE(I)=0
      NP=1
  307 READ(5,1000) NODNO,ZTYP(NP),NC(NP)
      IF(ABS(EOF(5)).NE.0.0) GO TO 309
      IF (NODNO.EQ.0) GO TO 309
  308 IF(NC(NP) .EQ. 0) NC(NP)=1
      NODE(NODNO)=NP
      NO=NC(NP)
      IF(ZTYP(NP) .NE. "WAVC,") GO TO 311
      IF(FLAG(1) .EQ. "CWIND") GO TO 315
      READ(5,1010) WV,DM,F
      IF (F.EQ.0.) GO TO 317
      GO TO 316
  315 READ(5,1010) DM,F
      IF (F.EQ.0.) GO TO 317
      WV = WNDVEL
  316 CALL WAVSIM(NP,WV,DM,F)
      ZTYP(NP)="WAVE,"
      GO TO 318
  317 HB(NP) = 0.
      ALPHAC(NP) = 0.
      DO 319 I=1,NO
  319 K(NP,I) = 9999.
      ZTYP(NP) = "SURF,"
      GO TO 321
  318 WRITE(6,2077) NODNO,WV,DM,F,A(NP,1),K(NP,1),W(NP,1)
      NP=NP+1
      GO TO 307
```

```
  311 CONTINUE                                                          SPECS     700
      IF(ZTYP(NP) .EQ. "WAVE,") READ(5,1010)((A(NP,I),K(NP,I),W(NP,I)),  SPECS     701
     $I=1,NO)                                                           SPECS     702
      IF(ZTYP(NP) .EQ. "WAVE,") WRITE(6,2080) NODNO, ((A(NP,I),          SPECS     703
     $K(NP,I), W(NP,I)),I=1,NO)                                         SPECS     704
      IF(ZTYP(NP) .EQ. "SURF,") READ(5,1020) HB(NP),ALPHAC(NP),(K(NP,I), SPECS     705
     $I=1,NO)                                                           SPECS     706
 321  IF(ZTYP(NP) .EQ. "SURF,") WRITE(6,2090) NODNO, HB(NP),             SPECS     707
     $ALPHAC(NP), (K(NP,I),I=1,NO)                                      SPECS     708
      NP=NP+1                                                           SPECS     709
      GO TO 307                                                         SPECS     710
312   IF (KKK.EQ.1) CALL RWIND (WFREQ,LW)                               SPECS     711
      AT = RAT+T*(KKK-1)                                                SPECS     712
      IF (ID.EQ.1 .AND. AMOD(AT,WFREQ) .LT. T) CALL RWAVE (AT,WFREQ)    SPECS     713
  309 RETURN                                                            SPECS     714
310   FORMAT (16A5/16A5)                                                SPECS     715
320   FORMAT (1H1)                                                      SPECS     716
325   FORMAT (2I5)                                                      SPECS     717
330   FORMAT (10X,16A5/10X,16A5)                                        SPECS     718
335   FORMAT(I5,3E10.3)                                                 SPECS     719
336   FORMAT(6E10.3)                                                    SPECS     720
341   FORMAT (12I5)                                                     SPECS     721
 348  FORMAT(5E10.3)                                                    SPECS     722
  349 FORMAT(2E10.3,T11,A10)                                            SPECS     723
350   FORMAT (7E10.3)                                                   SPECS     724
351   FORMAT (6E10.3,I5)                                                SPECS     725
  352 FORMAT(//,T31," SUSPENDED SEDIMENT    SCOUR      DEPOSITION",/,   SPECS     726
     $T12"SUBSTANCE  ELEMENT  SEDIMENT    FALL    CRIT. SHEAR CRIT. SHEA SPECS     727
     $R ERODABILITY",/,T15"ID",T23,"NUMBER   DIAMETER  VELOCITY    STRES SPECS     728
     $S       STRESS     COEFFICIENT",/)                                SPECS     729
  353 FORMAT(T15,I2,T24,I4,  T31,1PE10.3,T42,E10.3,T53,E10.3,T65,E10.3,T SPECS     730
     $77,E10.3)                                                         SPECS     731
  355 FORMAT(T15,I2,T25,"ALL"T31,1PE10.3,T42,E10.3,T53,E10.3,T65,E10.3,T SPECS     732
     $77,E10.3)                                                         SPECS     733
370   FORMAT (10X,"ERROR - TOO MANY NODES OR TOO MANY ELEMENTS")        SPECS     734
  380 FORMAT(10X,I5,T19,1PE10.3,1X,E10.3,1X,3(1X,E10.3))                SPECS     735
381   FORMAT(10X,6E10.3,I5//)                                           FETFIX1    25
 382  FORMAT(10X,5E10.3//)                                              SPECS     737
383   FORMAT(//,10X,"ALMBDA     AKP(1)      AKP(2)      AKP(3)     BETA1 FETFIX1    26
     1 AREA1      NODBET"/)                                             FETFIX1    27
  384 FORMAT(//,11X,"RHOSED(1) RHOSED(2) RHOSED(3)   RHOWAT  POROSITY  V SPECS     739
     $ISCOSITY",/)                                                      SPECS     740
  385 FORMAT(//,10X,"KD VALUES FOR SUBSTANCE",I2,/)                     SPECS     741
  386 FORMAT(10X,1P6E12.4)                                              SPECS     742
  390 FORMAT(//,10X,"CONNECTIVITY TABLE",//10X,"ELEMENT",T20,"NODES (IN SPECS     743
     $CCW SEQUENCE)",/)                                                 SPECS     744
400   FORMAT (//,10X,"NCOL = (BAND WIDTH + 1)/2 = ",I5)                 SPECS     745
410   FORMAT (10X,"DATA ERROR - BAND WIDTH TOO LARGE",//)               SPECS     746
  420 FORMAT(//,10X,"INITIAL CONDITION FOR EACH NODE",/)                SPECS     747
430   FORMAT (3E10.3)                                                   SPECS     748
 435  FORMAT(2I5,E10.3)                                                 SPECS     749
  440 FORMAT(//,10X,"VELOCITY AND DEPTH INPUT DATA",/)                  SPECS     750
  450 FORMAT(10X,I3,1X,1P3E11.4,1X,I3,1X,1P3E11.4,1X,I3,1X,1P3E11.4)    SPECS     751
  460 FORMAT(//,10X,"BOUNDARY CONDITIONS",/)                            SPECS     752
  470 FORMAT(10X,I5,5X,1PE15.5)                                         SPECS     753
  480 FORMAT(10X,"NODE",5X,"VX",9X,"VY",9X,"H",6X,"NODE",5X,"VX",9X,    SPECS     754
     1 "VY",9X,"H",6X,"NODE",5X,"VX",9X,"VY",9X,"H")                    SPECS     755
  490 FORMAT(10X," NODE         CONCENTRATION",/)                       SPECS     756
  500 FORMAT(I5,2E10.2,T1,A5)                                           SPECS     757
503   FORMAT(5E10.3)                                                    SPECS     758
505   FORMAT (E10.3)                                                    SPECS     759
509   FORMAT (I5)                                                       SPECS     760
```

```
510   FORMAT (6E10.3)
515   FORMAT(10X,"NONUNIFORM INITIAL CONDITIONS",11X,I5,/)
520   FORMAT (//)
 530  FORMAT(10X,6(I5,1PE15.7))
 531  FORMAT(14X,"NBED",5X,"BDIV",5X,"XYSO",5X,"BED",5X,"BD50",/)
532   FORMAT(9X,"M",1X,"NBED",5X,"BDIV",5X,"XYSO",5X,"BED",5X,"BD50",/)
533   FORMAT(7X,"J",5X,"GBA",7X,"GBB",7X,"GBC",7X,"GBD",7X,"GBE",7X,
     1"GBF",7X,"GBG",/)
 535  FORMAT(5X,2I5,4E10.3,//)
536   FORMAT(5X,I5,7E10.3)
 537  FORMAT(10X,I5,4E10.3,//)
538   FORMAT(10X,7E10.3,//)
539   FORMAT(11X,"GBA(1,1)  GBB(1,1)  GBC(1,1)  GBD(1,1)  GBE(1,1)  GBF(
     11,1)  GBG(1,1)",/)
540   FORMAT (10X,"DATA ERROR - NODE TABLE" )
550   FORMAT (10X,"TOTAL NUMBER OF CONSTITUENTS",12X,I5,/,10X,"NUMBER OF
     1TRIANGULAR ELEMENTS",13X,I4,/,10X,"NUMBER OF NODES",26X,I4,/,10X,"
     2NUMBER OF TIME PLANES",20X,I4,/,10X,"SIMULATION TIME STEP",24X,1PE
     310.4/10X,"PRINT FREQUENCY",25X,I5)
560   FORMAT (7I5)
570   FORMAT (10X,I5,2X,6I5)
  580 FORMAT(//,10X,"ELEMENT",T26,"X",T37,"Y",7X,"DECAY",
     $  5X,"SOURCE",5X,"ELEMENT",/,10X,"NUMBER",T21,"DISPERSION DISPERS
     $SION    TERM         TERM      THICKNESS",/)
  590 FORMAT(10X," NODE       BOUNDARY FLUX",/)
600   FORMAT (2(I5,E10.2))
  610 FORMAT(10X,"NODE",5X,"X-COORD",6X,"Y-COORD",/)
  611 FORMAT(10X,I3,3X,1PE11.4,2X,E11.4)
  705 FORMAT(10X,"NUMBER OF AREA SOURCES",18X,I5)
  706 FORMAT(/,13X,"CONSTITUENT",2X,"ELEMENT",3X,"SOURCE",/,15X,
     1   "NUMBER",5X,"NUMBER",3X,"STRENGTH")
  707 FORMAT(15X,I4,6X,I5,5X,1PE8.2)
  708 FORMAT(/,10X,"NUMBER OF POINT SOURCES",17X,I5)
  709 FORMAT(/,13X,"CONSTITUENT",3X,"NODE",5X,"SOURCE",/,15X,
     1   "NUMBER",5X,"NUMBER",3X,"STRENGTH")
 1000 FORMAT(I5,A5,I5)
 1010 FORMAT(3E10.4)
 1020 FORMAT(2F10.0/(5F10.0))
 1030 FORMAT(16A5)
 2000 FORMAT(1X,"SPECIFIED BOUNDARY CONDITION NODE",I5," IS OUT OF SEQUE
     $NCE.")
 2010 FORMAT(1X,"DERIVITIVE BOUNDARY CONDITION NODE",I5," IS OUT OF SEQU
     $ENCE.")
 2020 FORMAT(1H0,9X"STYP ="F5.2)
 2030 FORMAT(1H+,T23"EXPLICIT SOLUTION SELECTED")
 2040 FORMAT(1H+,T23"CRANK-NICHOLSON SOLUTION SELECTED")
 2050 FORMAT(1H+,T23"IMPLICIT SOLUTION SELECTED")
 2060 FORMAT(1X,"STYP NOT SPECIFIED WITHIN THE RANGE 0. TO 1.  -PROGRAM
     $TERMINATED")
 2070 FORMAT(10X"CONSTITUENT"I5,10X"FLAG FOR STANDARD FORM"I5)
 2075 FORMAT(///11X,"WAVE DATA INPUT"//11X"NODE"T19"WAVE ZONE"/)
 2077 FORMAT(10X,I5,T22,"WAVE",T28,"WIND VEL.=",E12.5,T51,"MEAN DEPTH=",
     $E12.5,T75,"EFFECTIVE FETCH LENGTH=",E12.5,/,T31," A=",E12.5,T46,"
     $K=",E12.5,T61," H=",E12.5,/)
 2080 FORMAT(10X,I5,T22"WAVE"T31" A="E12.5,T46" K="E12.5,T61" H="
     $E12.5,(/T34,E12.5,T33"="T48"="E12.5,T63"="E12.5))
 2090 FORMAT(10X,I5,T22"SURF"T31"H8="E12.5,T46" ALPHAC="F8.1,T61" K="
     $E12.5,(/T64,E12.5,T63"="))
 1025 FORMAT(12(A5,1X))
 6000 FORMAT(10X,"THE FOLLOWING OPTIONS HAVE BEEN ACTIVATED-",/)
 6015 FORMAT(15X,"NONUNIFORM COEFFICIENTS(DX, DY, ALFA, BETA, HS)")
 6020 FORMAT(15X,"NONUNIFORM VELOCITY AND/OR DEPTH.")
```

```
 6025 FORMAT(15X,"TIME VARYING BOUNDARY CONDITIONS.")                    SPECS     823
 6030 FORMAT(15X,"TIME VARYING HYDRAULICS.")                             SPECS     824
 6035 FORMAT(15X,"TIME VARYING INPUT DATA.")                             SPECS     825
 6040 FORMAT(15X,"UNIFORM SUSPENDED SEDIMENT DIAMETER.")                 SPECS     826
 6045 FORMAT(15X,"NONUNIFORM SUSPENDED SEDIMENT DIAMETER.")              SPECS     827
 6050 FORMAT(15X,"UNIFORM BED SEDIMENT DIAMETER.")                       SPECS     828
 6055 FORMAT(15X,"NONUNIFORM BED SEDIMENT DIAMETER.")                    SPECS     829
 6060 FORMAT(15X,"UNIFORM BED CONFIGURATION.")                           SPECS     830
 6065 FORMAT(15X,"NONUNIFORM BED CONFIGURATION.")                        SPECS     831
 6070 FORMAT(15X,"UNIFORM BED CONCENTRATION.")                           SPECS     832
 6075 FORMAT(15X,"NONUNIFORM BED CONCENTRATION.")                        SPECS     833
 6100 FORMAT(10X,"THE FOLLOWING OPTION HAS BEEN SELECTED FOR SUBSTANCE " SPECS     834
     $,I2,".",/)                                                         SPECS     835
 6110 FORMAT(15X,"NONUNIFORM INITIAL CONDITION.",//)                     SPECS     836
 6120 FORMAT(15X,"UNIFORM INITIAL CONDITION.")                           SPECS     837
 6125 FORMAT(15X,"SPATIALLY CONSTANT WIND")                              SPECS     838
 6126 FORMAT(F10.0)                                                      SPECS     839
 6127 FORMAT(18X,"OF MAGNITUDE",F5.1," M/SEC")                           SPECS     840
 6130 FORMAT(15X,"SPATIALLY VARYING WIND")                               SPECS     841
      END                                                                SPECS     842
      SUBROUTINE SURFTR(ID,H,UEXT,QT)                                    SURFTR      2
C                                                                        SURFTR      3
C**   THIS SUBROUTINE CALCULATES THE LITTORAL (LONGSHORE) TRANSPORT      SURFTR      4
C**   OF SEDIMENT IN THE SURF ZONE.                                      SURFTR      5
C**   BASED ON WORK DONE BY KOMAR                                        SURFTR      6
C                                                                        SURFTR      7
C     CHARACTER*5 ZTYP                                                   FETFIX1    28
      LOGICAL WAVE                                                       SURFTR      9
C                                                                        SURFTR     10
      COMMON/WAVE/ NODE(240), NC(135), A(135,10), K(135,10),             SURFTR     11
     $              W(135,10), HB(135), ALPHAC(135), WAVE, N, D          SURFTR     12
     $              , VIS, DPTHS(135)                                    SURFTR     13
      COMMON /ZTYPE/ ZTYP(135)                                           SURFTR     14
      COMMON/BLK12/ ACOF(6), U(6), V(6), DD(2), AKJ(9,100), ALMBDA,      SURFTR     15
     $              RHOWAT, AKP(3)                                       SURFTR     16
      COMMON/BLK15/ ILAYR(100,3), XYSO(100), BDIV(100), NBED(100),       SURFTR     17
     $              BED(100), RHOSED(3), XNT(100,3), RSAV1(100),         SURFTR     18
     $              RSAV2(100), RSAV3(100)                               SURFTR     19
      REAL K, IL                                                         SURFTR     20
C                                                                        SURFTR     21
C**   SLOPE   IS THE SLOPE OF THE BEACH.                                 SURFTR     22
C                                                                        SURFTR     23
      SLOPE = .012                                                       SURFTR     24
      PI=ACOS(-1.)                                                       SURFTR     25
      UEXT=UEXT/3600./24.                                                SURFTR     26
      APRIME=.6                                                          SURFTR     27
      G=9.8                                                              SURFTR     28
      SPWSED=RHOSED(ID)                                                  SURFTR     29
      SPWWAT=RHOWAT                                                      SURFTR     30
      RHOS=SPWSED/G                                                      SURFTR     31
      RHOW=SPWWAT/G                                                      SURFTR     32
      ALPHAB = ABS(ALPHAC(N))*PI/180.0                                   SURFTR     33
      C=SQRT(G*H)                                                        SURFTR     34
      EB=RHOW*G*HB(N)**2/8.                                              SURFTR     35
      UM=SQRT(2.*EB/(RHOW*H))                                            SURFTR     36
      ECNB=0.                                                            SURFTR     37
      T1=RHOW*G*HB(N)**2*C/16.                                           SURFTR     38
      NO=NC(N)                                                           SURFTR     39
      DO 10 I=1,NO                                                       SURFTR     40
C      WRITE (6,99) NO,N,I,K(N,I),H                                      SURFTR     41
C99    FORMAT (" NO, N, I, K(N,I), H = ",3I5,1P2E12.4)                   SURFTR     42
      ECNB=T1*(1.+(2.*K(N,I)*H/SINH(2.*K(N,I)*H)))+ECNB                  SURFTR     43
```

```
   10 CONTINUE
C
C**   CALCULATE THE IMMERSED WEIGHT LITTORAL TRANSPORT RATE, IL, IN
C**   KG(FORCE)/DAY.
C
      UBARL=2.7*UM*SIN(ALPHAB)*COS(ALPHAB) + UEXT
      COSALB = COS(ALPHAB)
      SINALB = SIN(ALPHAB)
C      WRITE (6,2020) SINALB,COSALB
C2020  FORMAT (" ?????? SIN(ALPHAB), COS(ALPHAB) = ",1P2E12.5)
      IL=.28*ECNB*UBARL/UM*3600.*24.
C
C**   NOW CALCULATE QT IN KG(FORCE)/DAY  , BASED ON THE SEDIMENT
C**   DRY WEIGHT.
C
      QT=IL*(RHOS/(RHOS-RHOW))
C
C**   CALCULATE QT PER UNIT WIDTH OF SURF ZONE.
C
      WW = DPTHS(N) / SLOPE *(H/DPTHS(N))
      QT = QT / WW
C       QT = QT / 150.
C      WRITE (6,1030) RHOS,RHOW,ALPHAB,EB,UM,T1
C1030  FORMAT ("***** RHOS,RHOW,ALPHAB,EB,UM,T1=",
C     1 /3X,6E12.4)
C      WRITE (6,1035) ECNB,UBARL,IL,WW,QT
C1035  FORMAT ("***** ECNB,UBARL,IL,WW,QT=",
C     1 /3X,5E12.4)
C      WRITE (6,1010) UEXT
C1010  FORMAT (" %%%%% UEXT=UEXY/3600./24.",F12.4)
      UEXT=UEXT*3600.*24.
C      WRITE (6,1020) UEXT
C1020  FORMAT (" %%%%% UEXT=UEXY*3600.*24.",F12.4)
      RETURN
      END
      SUBROUTINE TRANSP(NE,NROW,NCOL,T,ID,IDN,INO,K,NPRNT,NTP)
C
C     THIS ROUTINE CONSTRUCTS THE COEFFICIENT MATICES AND LOAD VECTOR
C     FOR THE SYSTEM OF ORDINARY DIFFERENTIAL EQUATIONS?
C             (P) (DY/DT) = (S) (Y) + (R)
C
      COMMON /BLK2/P(240,86),F(240)
      LEVEL  2,P,F
      COMMON /BLK3/S(240,86)
      LEVEL  2,S
      COMMON /BLK4/ R(240), RPAST(240,7), NODBET, BETA1, AREA1
      LEVEL  2,R,RPAST,NODBET,BETA1,AREA1
      COMMON /BLK6/VX(240),VY(240),H(240),STRESS(3)
      COMMON /BLK7/ NOD(240,6),X(240),Y(240)
      COMMON /BLK10/PEL(6,6),SEL(6,6),REL(6)
      COMMON /BLK11/D50(3,100),BD50(100),SR(3,100),SD(3,100)
      COMMON /BLK12/ACOF(6),U(6),V(6),D(2),AKJ(9,100),ALMBDA,RHOMAT,
     1            AKP(3)
      COMMON /BLK13/WS(3,100),CRSTRS(3,100),CDSTRS(3,100),ERODA(3,100)
      COMMON /BLK14/GBA(100,10),GBB(100,10),GBC(100,10),GBD(100,10),
     1             GBE(100,10),GBF(100,10),GBG(100,10),POR
      COMMON /BLK15/ILAYR(100,3),XYS0(100),BDIV(100),NBED(100),BED(100),
     1            RHOSED(3),XNT(100,3),RSAV1(100),RSAV2(100),RSAV3(100)
      COMMON/OPTION/ CLAYX, SANDX
C
      LOGICAL CLAYX, SANDX
C
```

```
C
      NBND=2*NCOL-1
      DO 100 I=1,NROW
      R(I)=0.
      DO 100 J=1,NBND
      P(I,J)=0.
      S(I,J)=0.
100   CONTINUE
C
      DO 103 M=1,NE
      AREA=0.0
      CALL ACOEFS(M,AREA)
      CALL PMATRX(M,AREA)
      IF (INO.LE.0) GO TO 111
      IF (ID.LE.3) CALL SHEAR(M)
      GO TO (104,105.106,107,108,108,108) ID
  104 IF(SANDX) CALL SAND2(M,ID,T,AREA)
      IF(CLAYX) CALL CLAY(M,ID,T)
      CALL SEDIME(M,AREA,ID)
      GO TO 110
105   CALL SILT(M,ID,T)
      CALL SEDIME(M,AREA,ID)
      GO TO 110
  106 IF(SANDX) CALL SILT(M,ID,T)
      IF(CLAYX) CALL SAND(M,ID,T,AREA)
      CALL SEDIME(M,AREA,ID)
      GO TO 110
107   CALL DISOLV(M,AREA,ID,IDN,T)
      GO TO 111
108   CALL PARTIC(M,AREA,ID,IDN,T)
      IF (ID.EQ.IDN) CALL BEDHIS(M,T)
      GO TO 111
  110 CONTINUE
C
C**   WRITE THE FOLLOWING INFORMATION FOR THE LAST 50 ODD
C**   NUMBERED TIME PLANES.
C
C       IF(M .EQ. 1) WRITE(6,200)
C       IF(M .EQ. 1) WRITE(6,205) K
C       WRITE(6,211) ID,M,STRESS(1),STRESS(2),STRESS(3),SR(ID,M),
C      1 SD(ID,M),ILAYR(M,ID)
111   CALL SMATRX (M,AREA,INO)
      IF(INO.LE.0.OR.KBC.GT.0) CALL RMATRX(M,AREA,ID,INO)
      CALL MATADD(M,NCOL,ID)
103   CONTINUE
C
C
C       IF(ID.GT.1)  GO TO 406
C       WRITE(6,403)
C 403   FORMAT(//,10X," SD MATRIX ",/)
C       WRITE(6,404)    (SD(1,I),I=1,NE)
C 404   FORMAT(10(1PE12.3))
C       WRITE(6,405)
C 405   FORMAT(//,10X," RSAV1 MATRIX ",/)
C       WRITE(6,404)  (RSAV1(I),I=1,NE)
C 406   CONTINUE
      RETURN
  200 FORMAT(//,10X,"ID ELEMENT",3X,"STRESS(1)",4X,"STRESS(2)",4X,
     1 "STRESS(3)",4X,"SR(ID,M)",4X,"SD(ID,M)",4X,"ILAYR(M,ID)")
  205 FORMAT(/,1X,T34,"FOR TIME SEGMENT NUMBER ",I4,/)
  211 FORMAT(10X,I2,3X,I3,3X,1PE11.4,2X,1PE11.4,2X,1PE11.4,2X,1PE11.4,
     1 2X,1PE11.4,6X,I4)
```

```
      END
      SUBROUTINE WAVSAN(ID,H,UEXT,QS)
C
C**   THIS SUBROUTINE CALCULATES THE SEDIMENT TRANSPORT
C**   DUE TO WAVES.  THE SOLUTION SCHEME IS BASED ON THE
C**   REPORT ≠SEDIMENT TRANSPORT IN RANDOM WAVES≠ BY S.S. LIANG AND
C**   HSIANG WANG, UNIVERSITY OF DELAWARE, DECEMBER, 1973.
C
C     CHARACTER*5 ZTYP
      LOGICAL WAVE
      COMMON/WAVE/ NODE(240),  NC(135), A(135,10), K(135,10),
     $             W(135,10), HB(135), ALPHAC(135), WAVE, N, D
     $             , VIS, DPTHS(135)
      COMMON /ZTYPE/ ZTYP(135)
      COMMON/BLK12/ ACOF(6), U(6), V(6), DD(2), AKJ(9,100), ALMBDA,
     $             RHOWAT, AKP(3)
      COMMON/BLK15/ ILAYR(100,3), XYSO(100), BDIV(100), NBED(100),
     $             BED(100), RHOSED(3), XNT(100,3), RSAV1(100),
     $             RSAV2(100), RSAV3(100)
      DIMENSION UZ(10), BETA(10), E(10), ANGLE(10), FCTR1(10),
     $             FCTR2(10), FCTR3(10), R(10), FCTR4(10)
      REAL K,NU
C
C**   CHECK TO SEE IF THE WAVE INFLUENCE WILL BE FELT
C**   BY THE BOTTOM.
C
      PI=ACOS(-1.)
      HBAR=0.
      NO=NC(N)
      DO 5 I=1,NO
    5 HBAR=2.*PI/K(N,I)+HBAR
      HBAR=HBAR/FLOAT(NO)
      IF(H .GT. HBAR/2.0) RETURN
C
C**   THE FOLLOWING SECTION INITIALIZES CONSTANTS
C
      UEXT=UEXT/3600./24.
      VIS=VIS/3600./24.
      SIGMA=5.15
      G=9.8
      AZ=587.
      OMEGA=1.
      BSTAR=4.
      ETAZ=1./1.5
      SPWSED=RHOSED(ID)
      SPWWAT=RHOWAT
      RHOS=SPWSED/G
      RHOW=SPWWAT/G
      NU=VIS
      B=18.*NU/((RHOS/RHOW+0.5)*D**2)
      Y=.35 * D
      GAMMA=3./(2.*RHOS/RHOW+1.)
      WV=-(RHOS/RHOW-1.)*G*D**2/(18.*NU)
      USQ=0.
C
C**   THE FOLLOWING SECTION INITIALIZES VARIABLES AND
C**   CALCULATES U SQUARED.
C
      DO 10 I=1,NO
      FCTR1(I)=0.
      FCTR2(I)=0.
      F1=0.
```

```
      EPSILON=ATAN(W(N,I)/B)
      BETA(I)=SQRT(W(N,I)/(2.*NU))
      E(I)=133.*SINH(K(N,I)*H)/(A(N,I)*BETA(I)*D)
      ANGLE(I)=0.6*BETA(I)*D
C
C**   CHECK IF EXPONENT IS TOO SMALL.
C
      IF((-2.*D*E(I)).GT. -675.) FCTR1(I)=EXP(-2.*D*E(I))
      IF((-4.*D*E(I)).GT. -675.) FCTR2(I)=EXP(-4.*D*E(I))
      FCTR3(I)=E(I)**2+.09*BETA(I)**2
      FCTR4(I)=E(I)**2-.09*BETA(I)**2
C
C**   FOR SHALLOW WATER CASE ONLY, THE FOLLOWING
C**   EXPRESSION FOR R(I) IS TRUE.
C
      R(I)=HV*H/(GAMMA*SIGMA*A(N,I)*W(N,I))
C
C**   CHECK IF EXPONENT IS TO SMALL.
C
      IF((-E(I)*Y) .GT. -675.) F1=0.5*EXP(-E(I)*Y)
      F2=0.3*BETA(I)*Y
      UZ(I)=A(N,I)*W(N,I)/SINH(K(N,I)*H)
C
C**   CALCULATE U SQUARED
C
      USQ= UZ(I)**2*(1.-2.*F1*COS(F2)+F1**2)+USQ
   10 CONTINUE
C
C**   NOW CALCULATE P AND C ZERO
C
      PSI=(RHOS-RHOW)*G*D/(RHOW*USQ)
      ARG=BSTAR*PSI-1./ETAZ
      CALL ERFC(ARG,P)
      P=.5*P
      GAMMAS=RHOS/RHOW
      CZBAR=AZ*P*GAMMAS
C
C**   NOW CALCULATE THE BEDLOAD TRANSPORT, QB
C**   QBE IS THE TRANSPORT DUE TO THE EXTERNAL VELOCITY, U.
C
      QB=0.
      T1=0.
      T2=0.
      T3=0.
      T4=0.
      DO 20 I=1,NO
      T1A=.5*K(N,I)*UZ(I)**2*FCTR1(I)
      T1A=T1A/(2.*NU*FCTR3(I)**2)
      T1B=2.*D*FCTR4(I)*SIN(ANGLE(I))
      T1C=1.2*E(I)*BETA(I)*D*COS(ANGLE(I))
      T1D=1./FCTR3(I)*(2.55*E(I)**2*BETA(I)-.0945*BETA(I)**3)*COS(ANGLE
     $(I))
      T1E=1./FCTR3(I)*(-.855*E(I)*BETA(I)**2+2.5*E(I)**3)*SIN(ANGLE(I))
C
      T1=T1A*(T1B+T1C+T1D+T1E)+T1
C
      T2A=0.5*K(N,I)*UZ(I)**2/(2.*NU*FCTR3(I))
      T2B=-.0375*BETA(I)/E(I)**2 *(FCTR2(I)-1.)
      T2C=.15*BETA(I)*D/E(I)
      T2D=2.4*BETA(I)*E(I)*D/FCTR3(I)
      T2E=1./FCTR3(I)**2
      T2E=T2E*(2.55*BETA(I)*E(I)**2-.0945*BETA(I)**3)
```

```
C
      T2=T2A*(T2B-T2C+T2D-T2E)+T2
C
      T3A=K(N,I)*UZ(I)**2/(2.*W(N,I))
      T3B=-D*FCTR1(I)*COS(ANGLE(I))
      T3C=0.5*E(I)/FCTR3(I)
      T3C=T3C*FCTR1(I)*COS(ANGLE(I))
      T3D=          0.15*BETA(I)/FCTR3(I)*FCTR1(I)*SIN(ANGLE(I))
      T3E=.25*FCTR2(I)/(2.*E(I))
      T3F=2.*D
      T3G=.5*E(I)/FCTR3(I)
      T3H=.25/(2.*E(I))
C
      T3=T3A*(T3B+T3C-T3D-T3E+T3F-T3G+T3H)+T3
C
      T4A=.5*K(N,I)*UZ(I)**2/(2*W(N,I)*FCTR3(I))
      T4B=0.5*E(I)*FCTR1(I)*COS(ANGLE(I))
      T4C=0.5*0.3*BETA(I)*FCTR1(I)*SIN(ANGLE(I))
      T4D=0.5/(2.*E(I))*FCTR4(I)*FCTR2(I)
      T4E=0.5*E(I)
      T4F=0.5/(2.*E(I))*FCTR4(I)
C
      T4=T4A*(T4B+T4C-T4D-T4E+T4F)+T4
C
   20 CONTINUE
C
      QBE=OMEGA*CZBAR*UEXT/(1.+1./7.)*(2.*D)**(1.+1./7.)*H**(-1./7.)
      QB=OMEGA*CZBAR*(T1+T2+T3+T4)+QBE
C
C
C**   CONVERT TO KG(FORCE)/DAY-M, 1PPM=1MG(FORCE)/LITER
C
      QB=QB/1000.*3600.*24.
C
C**   THE FOLLOWING SECTION CALCULATES THE SUSPENDED
C**   SEDIMENT TRANSPORT, QSUS.
C**   QSUSE IS THE SUSPENDED SEDIMENT TRANSPORT DUE TO THE EXTERNAL
C**   VELOCITY, UEXT.
C
      T1=0.
      T2=0.
      T4=0.
      DO 60 I=1,NO
      T1A=A(N,I)**2*W(N,I)*K(N,I)/(4.*(SINH(K(N,I)*H))**2)
      T1B=3.*SINH(2.*K(N,I)*H)/(2.*K(N,I)*H)
      Y=H
      FLAG=1.
      T1C=0.
C
C**   CHECK TO SEE IF DENOMINATOR IN T1CA IS ZERO
C
   21 IF ((R(I)+3.) .EQ. 0.) GO TO 25
      T1CA=(Y**(R(I)+3.))/(H**2*(R(I)+3.))
      GO TO 27
   25 T1CA=ALOG(Y)/(H**2)
C
C**   CHECK TO SEE IF DENOMINATOR IN T1CB IS ZERO
C
   27 IF ((R(I)+2.) .EQ. 0.) GO TO 28
      T1CB=(2*Y**(R(I)+2.))/(H*(R(I)+2.))
      GO TO 29
   28 T1CB=2./H*ALOG(Y)
```

```
   29 T1C=FLAG*(T1CA-T1CB)+T1C
      IF(FLAG .LE. 0.) GO TO 26
      Y=2.*D
      FLAG=-1.
      GO TO 21
C
   26 T1=T1A*T1B*T1C/((2.*D)**R(I))+T1
C
      T2A=UZ(I)**2*K(N,I)*.5/(2.*NU*FCTR3(I))
      T2B=1.2*BETA(I)*E(I)/FCTR3(I)
      T2C=0.075*BETA(I)/E(I)
      Y=H
      FLAG=1.
      T3A=0.
      T3B=0.
      T3C=0.
C
C**   CHECK TO SEE IF DENOMINATOR OF T3A IS ZERO
C
   41 IF((R(I)+3.) .EQ. 0.) GO TO 43
      T3A=FLAG*1.5*Y**(R(I)+3.)/(H**2*(R(I)+3.))+T3A
      GO TO 44
   43 T3A=FLAG*1.5/(H**2)*ALOG(Y)+T3A
C
C**   CHECK TO SEE IF DENOMINATOR OF T3B IS ZERO
C
   44 IF((R(I)+2.) .EQ. 0.) GO TO 45
      T3B=FLAG*3.*Y**(R(I)+2.)/(H*(R(I)+2.))+T3B
      GO TO 46
   45 T3B=FLAG*3./H*ALOG(Y)+T3B
C
C**   CHECK TO SEE IF DENOMINATOR OF T3C IS ZERO
C
   46 IF((R(I)+1. ) .EQ. 0.) GO TO 47
      T3C=FLAG*Y**(R(I)+1.)/(R(I)+1.)+T3C
      GO TO 48
   47 T3C=FLAG*ALOG(Y)+T3C
   48 IF (FLAG .LE. 0.) GO TO 42
      Y=2.*D
      FLAG=-1.
      GO TO 41
C
   42 T3=(T3A-T3B+T3C)/((2.*D)**R(I))
C
      T2=T2A*(T2B-T2C)*T3+T2
C
C**   THIS SECTION CALCULATES PART OF THE VALUES(T4) REQUIRED TO
C**   CALCULATE QSUSE.
C
      IF(UEXT .EQ. 0.) GO TO 60
      Y=H
      FLAG=1.
      T4A=(2.*D)**(-R(I))
      T4B=0.
C
C**   CHECK TO SEE IF DENOMINATOR OF T4B IS ZERO.
C
   51 IF((1.+1./7.+R(I)) .EQ. 0.) GO TO 53
      T4B=FLAG*Y**(1.+1./7.+R(I))/(1.+1./7.+R(I))+T4B
      GO TO 55
   53 T4B=FLAG*ALOG(Y)+T4B
   55 IF(FLAG .LE. 0.) GO TO 57
```

```
      Y=2.*D
      FLAG=-1.
      GO TO 51
   57 T4=T4A*T4B+T4
   60 CONTINUE
C
      QSUSE=CZBAR*UEXT/H**(1./7.)*T4
      QSUS=CZBAR*(T1+T2)+QSUSE
C
C**   CONVERT TO KG(FORCE)/DAY-M
C
      QSUS=QSUS/1000.*3600.*24.
C
C**   NOW CALCULATE THE TOTAL SEDIMENT TRANSPORT
C
      QS=QB+QSUS
C     WRITE (6,1500) QS,QB
C1500 FORMAT (" $$$$$ QS, QB = "1P2E12.4)
      UEXT=UEXT*3600.*24.
      VIS=VIS*3600.*24.
      RETURN
      END
      SUBROUTINE WAVSIM(NP,V,H,F)
C
      LOGICAL WAVE
      COMMON/WAVE/ NODE(240),  NC(135), A(135,10), K(135,10),
     $              W(135,10), HB(135), ALPHAC(135), WAVE, N, D, VIS
     $             ,DPTHS(135)
      COMMON /ZTYPE/ ZTYP(135)
C     CHARACTER*5 ZTYP
      REAL K
      I=1
C
C**   CHECK FOR V EQUAL 0.  THE WAVE GENERATOR WILL
C**   NOT ALLOW V TO BE IDENTICALLY 0.
C
      IF(V .EQ. 0.) V=.01
C
C**   CONVERT TO PROPER UNITS FOR HS AND TS EQ.
C
      V=V*3.2808/(24.*3600.)
      H=H*3.2808
      F=F*3.2808
      G=32.2
      PI=ACOS(-1.)
      T1=G*H/(V**2)
      T2=G*F/(V**2)
C
C**   CALCULATE THE SIGNIFICANT WAVE HEIGHT
C
      HS=.283*V**2/G*TANH(.530*T1**.75)*
     $TANH((.0125*T2**.42)/TANH(.530*T1**.75))
C
C**   USE THE RMS WAVE HEIGHT
C
      HRMS=HS/1.416
C
C**   CONVERT TO METRIC UNITS
C
      HRMS=HRMS/3.2808
      A(NP,I)=HRMS/2.
C
```

```
C** CALCULATE THE SIGNIFICANT WAVE PERIOD
C
      TS=1.2*2.*PI*V/G*TANH(.833*T1**.375)*
     $TANH((.077*T2**.25)/TANH(.833*T1**.375))
      W(NP,I)=2.*PI/TS
      G=9.8
      V=V/3.2808*24.*3600.
      H=H/3.2808
      F=F/3.2808
      K(NP,I)=SOL(H,G,W(NP,I),0.0001)
      RETURN
      END
```

DISTRIBUTION

No. of
Copies

OFFSITE

U.S. Nuclear Regulatory
Commission
Division of Technical
Information and
Document Control
7920 Norfolk Avenue
Bethesda, MD 20014

P. R. Reed
U.S. Nuclear Regulatory
Commission
Washington, DC 20555

R. B. Codell
U.S. Nuclear Regulatory
Commission
Washington, DC 20555

F. Swanberg
Office of Nuclear
Regulatory Research
U.S. Nuclear Regulatory
Commission
Washington, DC 20555

E. F. Conti
Office of Nuclear
Regulatory Research
U.S. Nuclear Regulatory
Commission
Washington, DC 20555

C. V. Alonso
U.S. Department of Agriculture
Sedimentation Laboratory
Southern Region
Agriculture Research Center
P.O. Box 1157
Oxford, MI 38655

No. of
Copies

ONSITE

50 Pacific Northwest Laboratory

D. B. Cearlock
D. W. Dragnich
P. C. Hays
E. A. Jenne
Y. Onishi (32)
W. L. Templeton
F. L. Thompson (4)
D. S. Trent
L. L. Wendell
Technical Information (5)
Publishing Coordination (2)

NRC FORM 335 (11-81)

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

1. REPORT NUMBER (Assigned by DDC): NUREG/CR-2424, Vol. 2; PNL-5088

4. TITLE AND SUBTITLE (Add Volume No., if appropriate): Volume 2: Users Manual and Computer Program Listing for FETRA

2. (Leave blank)

3. RECIPIENT'S ACCESSION NO.

7. AUTHOR(S): Y. Onishi, F. L. Thompson

5. DATE REPORT COMPLETED: MONTH April, YEAR 1984

DATE REPORT ISSUED: MONTH May, YEAR 1984

9. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code):
Pacific Northwest Laboratory
P.O. Box 999
Richland, WA 99352

6. (Leave blank)

8. (Leave blank)

12. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code):
Division of Health, Siting and Waste Management
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC

10. PROJECT/TASK/WORK UNIT NO.

11. FIN NO.: B2271

13. TYPE OF REPORT: Topical

PERIOD COVERED (Inclusive dates)

15. SUPPLEMENTARY NOTES

14. (Leave blank)

16. ABSTRACT (200 words or less)

FETRA is a finite element model for simulating the sediment and contaminant transport to surface water. The model was applied to a test site in the Irish Sea and modified to account for wave mechanisms that affect sediment suspension. Volume 2 of this report presents a very brief users guide for FETRA and a computer program listing of the model.

17. KEY WORDS AND DOCUMENT ANALYSIS

17a. DESCRIPTORS

Computer Code for Ocean/Marine Environment
Sediment/Radionuclide
Transport Model
FETRA

17b. IDENTIFIERS/OPEN-ENDED TERMS

18. AVAILABILITY STATEMENT: Unlimited

19. SECURITY CLASS (This report): Unclassified

20. SECURITY CLASS (This page)

21. NO. OF PAGES

22. PRICE: $

NRC FORM 335 (11-81)

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

FOURTH CLASS MAIL
POSTAGE & FEES PAID
USNRC
WASH. D. C.
PERMIT No. G 67

N11389-00003 2
USNRC
JO ANN RESNER
W 501

NUREG/CR-2424, Vol. 2

MATHEMATICAL SIMULATION OF SEDIMENT AND RADIONUCLIDE TRANSPORT IN COASTAL WATERS

MAY 1984