United States
Environmental Protection
Agency

Environmental Research
Laboratory
Athens GA 30613

EPA-600 3-82-055
May 1982

Research and Development

EPA

User's Manual for the Instream Sediment-Contaminant Transport Model SERATRA

EPA 600/3-82-055
May 1982

USER'S MANUAL FOR THE INSTREAM SEDIMENT-CONTAMINANT TRANSPORT MODEL SERATRA

by

Y. Onishi and S.E. Wise
Battelle
Pacific Northwest Laboratories
Richland, Washington 99352

Contract No. 68-03-2613

Project Officer

Robert B. Ambrose
Technology Development and Applications Branch
Environmental Research Laboratory
Athens, Georgia 30613

ENVIRONMENTAL RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
ATHENS, GEORGIA 30613

NOTICE

Mention of trade names or commercial products does not consititute endorsement or recommendation for use.

FOREWORD

As environmental controls become more costly to implement and the penalties of judgment errors become more severe, environmental quality management requires more efficient management tools based on greater knowledge of the environmental phenomena to be managed. As part of this Laborarory's research on the occurrence, movement, transformation, impact, and control of environmental contaminants, the Technology Development and Applications Branch develops management and engineering tools to help pollution control officials achieve water quality goals through watershed management.

Many toxic contaminants are persistent and undergo complex interactions in the environment. As an aid to environmental decision-makers, the Chemical Migration and Risk Assessment methodology was developed to predict the occurrence and duration of pesticide concentrations in surface waters receiving runoff from agricultural lands and to assess potential acute and chronic damages to aquatic biota.

David W. Duttweiler
Director
Environmental Research Laboratory
Athens, Georgia

ABSTRACT

This manual guides the user in applying the sediment-contaminant transport model SERATRA. SERATRA is an unsteady, two-dimensional code that uses the finite element computation method with the Galerkin weighted residual technique. The model has general convection-diffusion equations with decay and sink/source terms with appropriate boundary conditions. A sediment transport submodel, a dissolved contaminant transport submodel, and a particulate contaminant (contaminants adsorbed by sediment) transport submodel are coupled to include the effects of sediment-contaminant interaction.

SERATRA is an integral part of the Chemical Migration and Risk Assessment Methodology, which predicts overland and instream pesticide migration and fate to assess the potential short- and long-term impacts on aquatic biota in receiving streams. Companion reports to this document are Methodology for Overland and Instream Migration and Risk Assessment of Pesticides, Mathematical Model SERATRA for Sediment-Contaminant Transport in Rivers and Its Application to Pesticide Transport in Four Mile and Wolf Creeks in Iowa, User's Manual for EXPLORE-I: A River Basin Water Quality Model (Hydraulic Module Only), and Frequency Analysis of Pesticide Concentrations for Risk Assessment (FRANCO Model).

This report was submitted in partial fulfillment of Contract No. 68-03-2613 by Battelle Pacific Northwest Laboratories under the sponsorship of the U.S. Environmental Protection Agency. This report covers the period April 1978 to January 1980, and work was completed as of January 1980.

CONTENTS

SECTION 1

INTRODUCTION

This manual is the guide to the sediment-contaminant transport model, SERATRA (Onishi et al. 1976; Onishi, 1977, Onishi et al. 1979a). SERATRA is an unsteady, two-dimensional code which utilizes the finite element computation method with the Galerkin weighted residual technique. The model has general convection-diffusion equations with decay and sink/source terms with appropriate boundary conditions. It consists of three submodels coupled to include the effects of sediment-contaminant interaction. The submodels are:

1. a sediment transport submodel
2. a dissolved contaminant transport submodel
3. a particulate contaminant (contaminants adsorbed by sediment) transport submodel.

SERATRA was used as an integral part of the Chemical Migration and Risk Assessment (CMRA) Methodology to predict overland and instream pesticide migration and fate and to assess the potential short- and long-term pesticide impacts on aquatic biota in receiving streams. Separate reports describe the mathematical model formulation and application results of SERATRA (Onishi and Wise, 1979) and the overall CMRA Methodology (Onishi et al. 1979b).

The model consists of one main program and 41 subroutines. This manual provides detailed program structures, input instructions, explanations of required input data, samples of input data and instructions for computer program operations.

SECTION 2

PROGRAM DESCRIPTIONS

SERATRA is written in the FORTRAN preprocessor language, FLECS. However, a standard Fortran IV version of SERATRA is also available.

The SERATRA source package contains all of the program FLECS source files, a labeled COMMON include file, a size parameter include file, and an overlay descriptor file. Also included are various indirect command files that are used to create FORTRAN sources, compile programs, list the program sources, and create the task image. The purpose of this chapter is to describe each file or subprogram and to explain the sequence of events and data file actions that occur during a simulation.

COMPUTATION PROCEDURE

The simulation procedure consists of the following five major steps:

1. Read the problem definitions.
2. Read and process input data for each river segment.
3. Prepare temporary data and the segment's result file.
4. Solve the sediment and contaminant transport equations.
5. Write the simulation results to result files.

Step 1 is only performed once. Steps 2 through 5 are repeated for each river segment being simulated and Steps 4 and 5 are repeated for each time step in the simulation. This procedure could be described as a "marching solution" because each river segment is modeled for all time steps before moving downstream to the next river segment.

Each major step is made up of many sub-steps. Detailed description of each major step is as follows:

1. Read the problem definitions

   a. Read the input file name and result file name from the console input device.
   b. Read the simulation control variables, number of segments, simulation length, and other parameters that describe the general characteristics of the simulation from the input file.
   c. Open the time series analysis data file.

2. Read and process the data for each river segment

a. Read the river segment dimensions.
b. Read the sediment characteristics.
c. When processing the data for the first segment, read the photolysis data.
d. Read initial bed conditions of sediment and contaminants in the segment.
e. Read initial water column conditions of sediment and contaminant in the segment.
f. When processing Segment 1 (upstream-end segment) read sediment and contaminant concentrations from upstream of the segment.
g. Read concentrations of sediment and contaminants from tributaries, if any.
h. Read the hydrologic data.

3. Prepare temporary data and the segment's result file.

a. Convert initial nodal concentrations to cell-centered concentrations.
b. Build the segment's result file name and open the file for output.

4. Solve the sediment and contaminant transport equations.

a. Update the flow and concentration values for the new time step.
b. Redistribute the flows and concentrations if the water depth within the segment has changed between time steps.
c. Add any concentration contributions from a tributary source to the concentrations from upstream.
d. Calculate the amount of deposition to the river bed or resuspension from the bed for sand, silt, and clay.
e. Determine the coefficients of convection, diffusion, decay, and source terms in the sediment and particulate contaminant transport convection-diffusion equations for each of three sediment size fractions and in the dissolved contaminant transport convection-diffusion equation.
f. Solve the sediment transport equations to compute vertical distribution of sediment for each sediment size fraction.
g. Solve the particulate contaminant transport equations to compute vertical distribution of contaminant attached to sediment of each size fraction.
h. Solve dissolved contaminant transport equations to compute vertical distributions of dissolved contaminant.
i. Redistribute the sediment and particulate contaminant concentrations due to the settling velocities of the sediments.
j. Calculate the decay of the contaminant in the river bed.
k. Update the bed history with new deposition and resuspension of sediments and particulate contaminants.

5. Write the simulation results to result files

a. Save computed concentrations which will become input to the segment downstream.

b. Save the concentrations to segment's result files for post-processing by Program SPPR.
c. Write the depth-averaged concentrations to the time series analysis file for post-processing by Program FRANCO (Onishi et al, 1979b).

PROGRAM MODULE DESCRIPTION

Figure 1 is a schematic of the model linkage. Following is the description of each program module in alphabetical order following the main program.

Executive Program - SERATRA

Except for some concentration conversions and averaging, the executive program's main responsibility is calling the other subprograms. It does, however, write the results at each time step to the time series analysis file, the segment's result file, and the next segment's upstream condition input file. It also controls the segment and time-step loops.

The executive program calls the following subroutines:

STRTUP	WTRDAT	FCODE	SAVEIT
INIDAT	UPSDAT	HYDFLO	DIAG
DIMDAT	TRBDAT	ICFLO	
SEDDAT	HYDDAT	RDSFLO	
PHOINP	RPTERR	TRBFLO	
BEDDAT	COLLAP	TRANSP	

Subroutine BEDDAT

Called at the beginning of simulation in each segment to read the segment's initial bed conditions that include the sediment weight fractions of three sediment sizes, and contaminant concentrations associated with these three sediment size fractions.

Called by SERATRA

Subroutine BEDDK

Computes changes in the bed contaminant concentrations caused by radionuclide decay.

Called by TRANSP

Subroutine BEDHIS

Maintains and updates a record of bed history for river-bed elevation, bottom sediment size fraction ratio, and associated contaminant concentrations.

Called by TRANSP

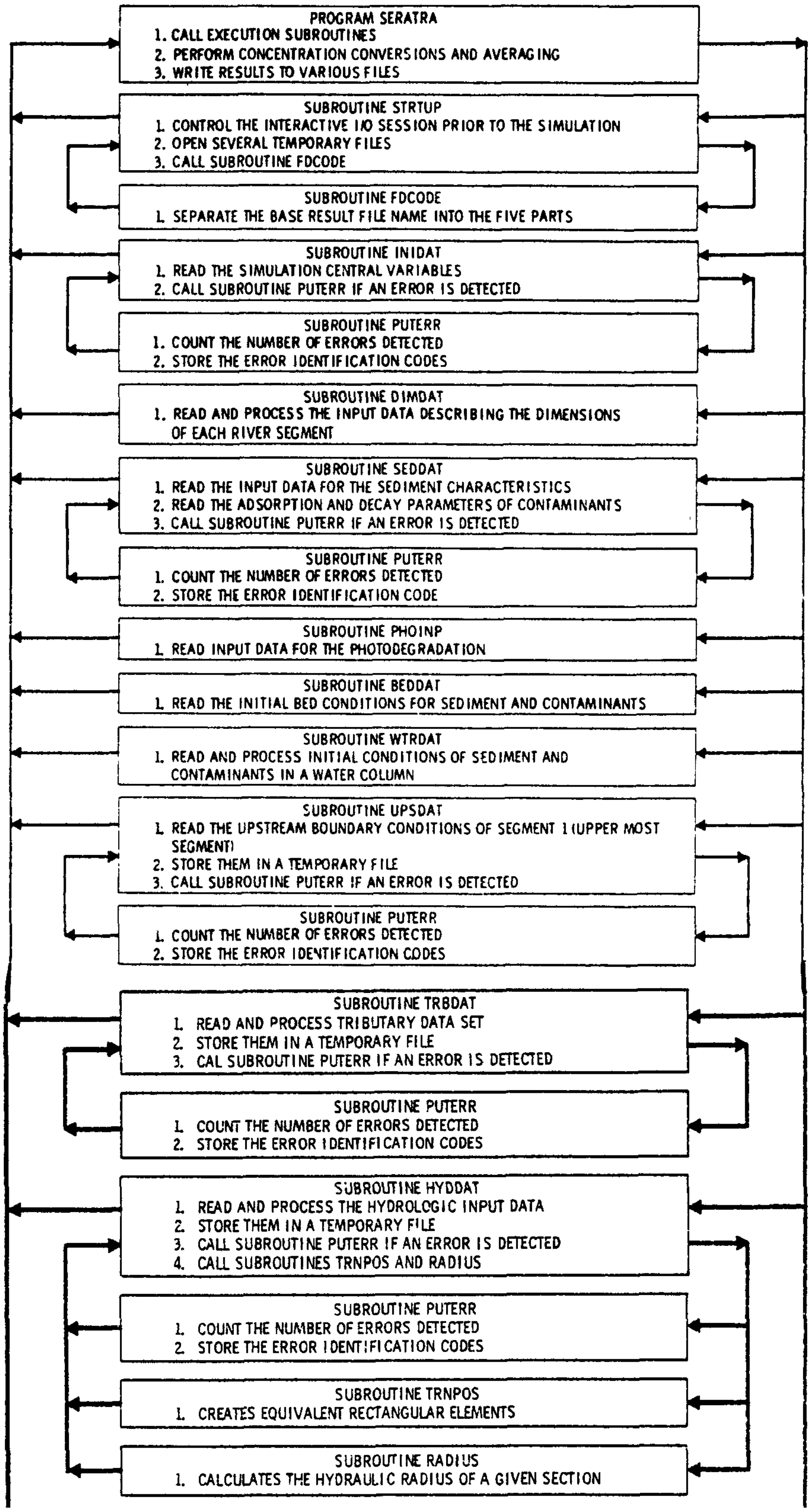


Figure 1. Schematic of Model Linkage

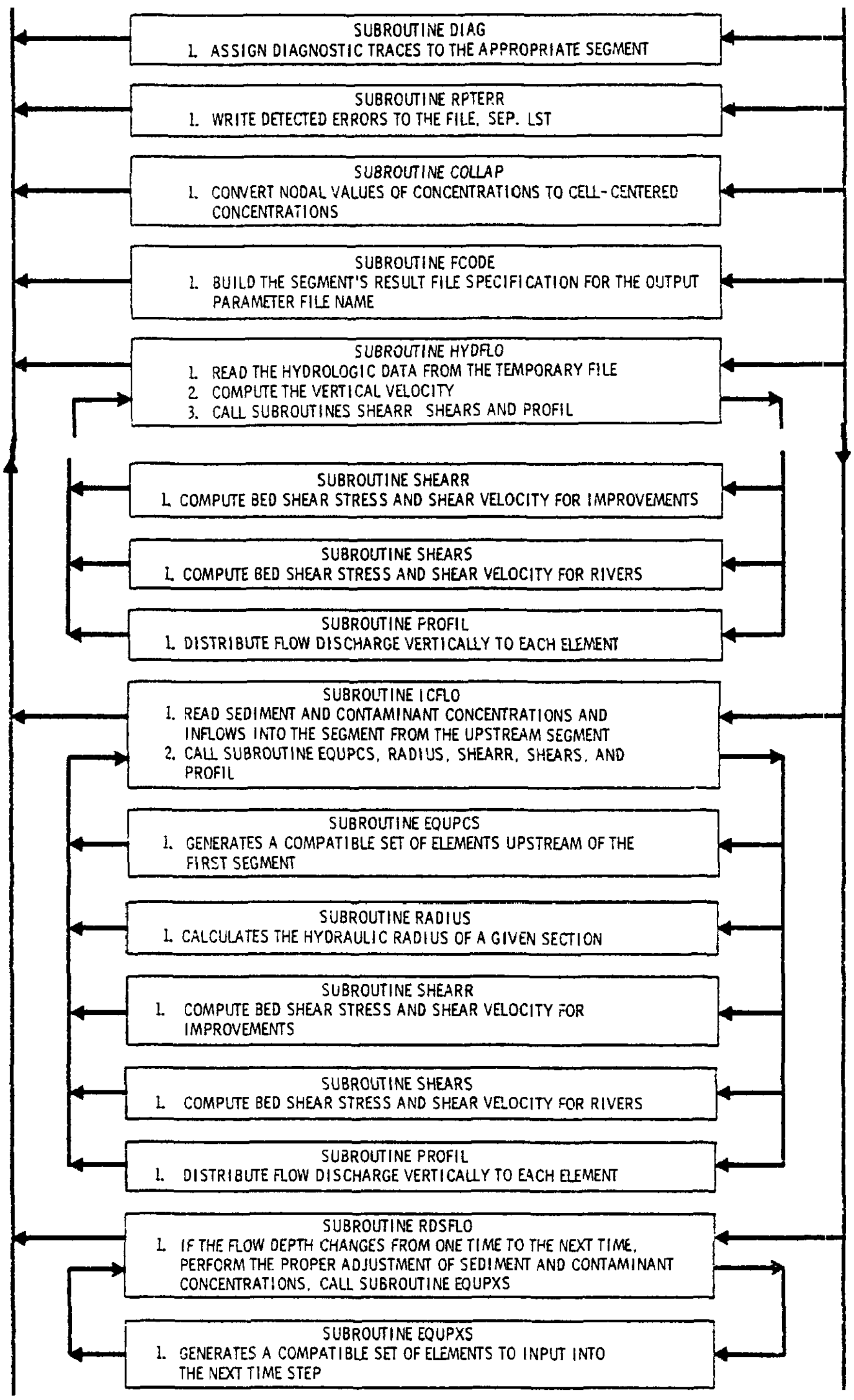


Figure 1. (contd)

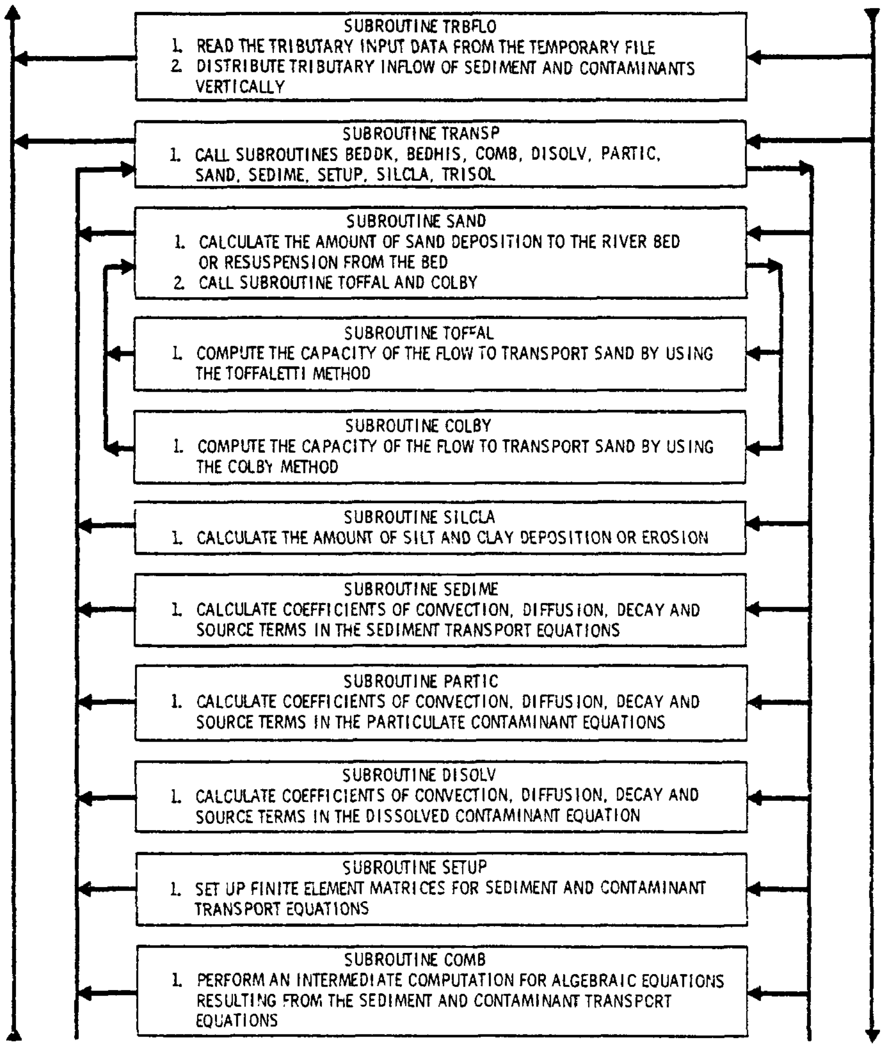


Figure 1. (contd)

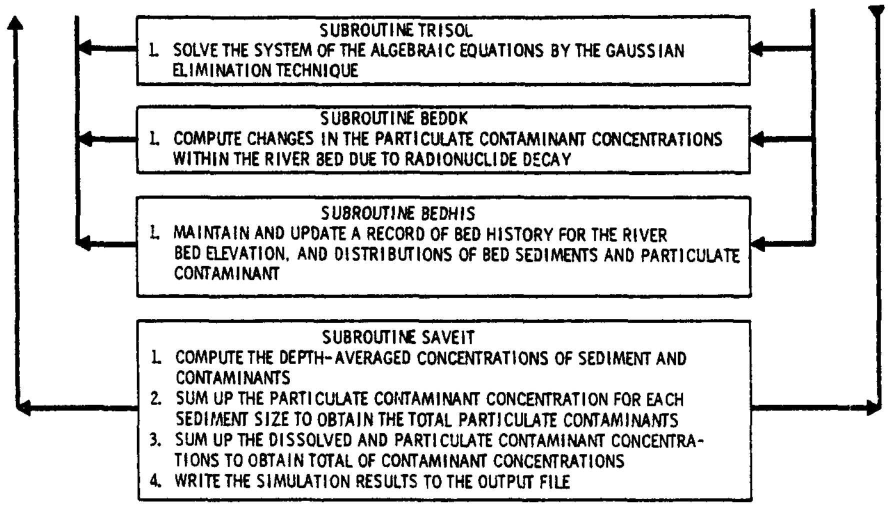


Figure 1. (contd)

Subroutine BEDDAT

Reads and processes initial bed conditions.

Called by SERATRA

Subroutine BEDDK

Calculates the decay of the contaminants in the river bed.

Called by TRANSP

Subroutine BEDHIS

Adjusts bed surface elevations, sediment fractions, and associated concentrations due to scouring and deposition.

Called by TRANSP

Subroutine COLBY

The model, SERATRA, offers options to select either the Colby or Toffaletti methods to compute the capacity of the flow to transport sand in rivers. This subroutine describes the Colby method. The Colby method includes the effects of temperature and fine sediment concentration on sand transport explicitly and is generally applied to small streams.

Called by SAND

Subroutine COLLAP

Converts nodal values of concentrations to cell-centered concentrations.

Called by SERATRA

Subroutine COMB

For the following equation:

$$[P] \{C\}^{n+1} = [S] \{C\}^{n} + \{R\},$$

this subroutine calculates the right hand side of the equation.

Called by TRANSP

Subroutine DIMDAT

Reads and processes the input data that describe the dimensions of each segment. Dimensions include segment length, surface area and thickness of each vertical water column element, bed layer thicknesses, initial bed thickness, numbers of bed layers and vertical water column elements, bed surface elevation, and porosity of bed sediment.

Called by SERATRA.

Subroutine DISOLV

Calculates the coefficients of convection, diffusion, decay and source terms in the dissolved contaminant transport equation.

Called by TRANSP

Subroutine EQUPCS

Calculates depth of upstream elements to match conditions at segment 1.

Called by ICFLO

Subroutine EQUPXS

Calculates depth of elements previous to the present computational segment.

Called by RDSFLO

Subroutine FCODE

Builds the segment's result file specification for the output parameter file name from the device name, group identification code, user identification code, base file name, extension and the segment number.

Called by SERATRA

Subroutine FDCODE

Separates the base result file name into the five parts of device, group identification code, user identification code, base file name, and extension.

Called by STRTUP

Subroutine HYDDAT

Reads and processes the hydrologic input data. The data are read (logical Unit 1), checked for consistency, and written to the temporary file HYDROLOGY.TMP (logical Unit 4).

Called by SERATRA
Calls TRNPOS, RADIUS, and PUTERR

Subroutine HYDFLO

Called each time step to read the hydrologic data from the file HYDROLOGY.TMP, which has been prepared by subroutine HYDDAT. It also computes the vertical velocities, if any.

Called by SERATRA
Calls SHEARR, SHEARS, and PROFIL

Subroutine ICFLO

Reads the sediment and contaminant concentrations and inflows into the segment from the upstream segment. For Segment 1, the data are read from the file prepared by Subroutine UPSDAT. For the other segments, the data are the simulation results for the stream segment that were saved by the executive routine.

Also redistributes sediment and contaminant concentrations within the segment by calling Subroutine SEGMTX if the depth of the upstream segment differs from the depth of the current segment.

Called by SERATRA
Calls EQUPCS, RADIUS, SHEARR, SHEARS, PROFIL

Subroutine INIDAT

Reads the simulation control variables such as number of time steps, number of segments, time step save frequency, time step length, and the time series analysis control parameters.

Called SERATRA
Calls PUTERR

Subroutine PARTIC

Calculates the coefficients of convection, diffusion, decay, and source terms in the particulate contaminant transport equation. This calculation is performed for the particulate contaminant attached to each sediment in three sediment size fractions.

Called by TRANSP

Subroutine PHOINP

Reads data and parameters for the dissolved contaminant degradation due to photolysis.

Called by SERATRA

Subroutine PROFIL

Assigns the flow discharge to each element according to a logarithmic or uniform vertical velocity profile from the cross-sectionally averaged discharge. The decision of which profile to be used is based upon the distance from the river bottom to the elevation where the logarithmic velocity becomes theoretically zero. When this distance is greater than one fourth the bottom element thickness, a uniform profile is used. Otherwise, the logarithmic profile is assigned to the flows.

Called by HYDFLO, ICFLO

Subroutine PUTERR

Called when an error is detected during the input data processing. PUTERR stores the error identification code and counts a number of error detected.

Called by DIMDAT, INIDAT, TRBDAT, and UPSDAT

Subroutine RADIUS

Calculates the hydraulic radius for a given depth and geometry.

Called by HYDDAT, ICFLO

Subroutine RDSFLO

Redistributes the sediment and contaminant concentrations within the same segment whenever the flow depth changes from one time to the next.

Called by SERATRA
Calls EQUPXS

Subroutine RPTERR

If any errors are detected during the input of data processing, this subroutine writes these errors to the file SED.LST (logical unit 6). It opens the error message file SERATERR.MSG (logical unit 10), reads the appropriate error messages, and reports them.

Called by SERATRA

Subroutine SAND

Calculates the amount of sand deposition to the river bed or resuspension from the bed. Availability of sand to be scoured from the river bed is also checked if resuspension occurs. The user decides whether to use the Colby or Toffaletti sediment discharge formula to compute the capacity of the flow to transport the noncohesive sediment for an assigned river reach. However, if the Colby method is selected and should fail during the simulation period, as a result of some limitations, the Toffaletti method automatically replaces the Colby method to continue the simulation.

Called by TRANS
Calls TOFFAL and COLBY

Subroutine SAVEIT

Writes the simulation results to the binary output file (logical unit 5) at predetermined time steps. SAVEIT computes the depth-averaged concentrations of sediment and contaminants, total particulate concentration (sum of depth-averaged particulate contaminants associated with three sediment size fractions), total contaminant concentrations (sum of dissolved and particulate contaminants), and average contaminant concentrations for each river bed layer before they are written to the file.

Called by SERATRA

Subroutine SEDDAT

Reads the input data related to sediment characteristics for each river segment. These values include sediment fall velocities, densities, diameters, critical shear stress values, vertical diffusion coefficients, contaminant adsorption values, and decay parameters. It computes rates of contaminant decay due to hydrolysis, oxidation, and biodegradation.

Called by SERATRA
Calls PUTERR

Subroutine SEDIME

Calculates the coefficients of convection, diffusion, decay, and source terms in the sediment transport equations for each sediment size fraction.

Called by TRANS

Subroutine SETUP

Sets up finite-element matrices [P], [S] and [R] for sediment and contaminant transport.

$$[P] \left\{\frac{dc}{dt}\right\} + [S]\{C\} = \{R\}$$

Called by TRANS

Subroutine SHEARR

Computes bed shear stress and shear velocity for the sediment-laden flow in impoundments such as reservoirs where the slope of the bed is not parallel to the slope of the water surface. Computations are based on the velocity, depth, and bed-sediment diameter.

Called by HYDFLO

Subroutine SHEARS

Computes the bed shear stress and shear velocity of free-flowing water bodies, such as rivers and streams. The bed slope, hydraulic radius and specific weight of water are the basis for the calculations. This method is not applicable to areas of impoundments.

Called by HYDFLO

Subroutine SILCLA

Calculates the amount of silt and clay deposition or erosion. Availability of silt and clay on the river bed is checked if erosion occurs. The Partheniades and Krone formulas are used to compute the rates of cohesive sediment deposition or erosion.

Called by TRANSP

Subroutine STRTUP

Responsible for controlling the interactive I/O session prior to the simulation and for opening the following temporary files:

Name	LUN	Contents
DUMMY.DT1	2	Inflow/outflow for Segment 1, Initial conditions to Segment 1
DUMMY.DT2	3	Inflow/outflow for other segments
HYDROLOGY.TMP	4	Current segment's hydrologic data
SED.LST	6	Line printer listing file
TRIBUTARY.TMP	7	Current segment's tributary data

Called by SERATRA
Calls FDCODE

Subroutine TOFFAL

User can choose either the Toffaletti or Colby methods to compute the capacity of the flow to transport sand. This routine is used to compute sediment capacity by the Toffaletti method.

Called by SAND

Subroutine TRANSP

Controls Step 4 (solve the sediment and contaminant transport equations) as described in the previous section. The major task of this subroutine is to call other subroutines.

Called by SERATRA
Calls BEDDK, BEDHIS, COLLAP, COMB, DISOLV, PARTIC, SAND, SEDIME, SETUP, SILCA and TRISOL

Subroutine TRBDAT

Reads and processes tributary data set. The data are read from the input stream (logical unit 1), and written to a temporary file, TRIBUTARY.TMP (logical unit 7). Tributary data include sediment concentration for each sediment size fraction, concentration of particulate contaminant associated with each sediment size fractions, and concentration of dissolved contaminant.

Called by SERATRA
Calls PUTERR

Subroutine TRBFLO

If a segment contains a confluence with a tributary, this routine is called each time step to read the tributary input data from TRIBUTARY.TMP (logical unit 7) and to redistribute concentrations to each vertical layer.

Called by SERATRA

Subroutine TRISOL

Solves the system of equations [A] {C} = [B] for {C}. The Gaussian elimination technique is used in this program.

Called by TRANSP

Subroutine TRNPOS

Creates idealized rectangular geometry from input cross-sections.

Called by HYDDAT

Subroutine UPSDAT

Responsible for reading the upstream inflow conditions to Segment 1 (upper most segment) from the input stream (logical unit 1) and writing them to DUMMY.DT1 (logical unit 2) for subsequent use during the simulation of Segment 1.

Called by SERATRA
Calls PUTERR

Subroutine WTRDAT

Reads and processes initial sediment and contaminant concentrations in segments except Segment 1 prior to the simulation. Data to be read from the input stream (logical unit 1) include the sediment concentration for each sediment size fraction, the concentrations of the contaminant attached to a sediment of each size fraction, and dissolved contaminant concentration.

Called by SERATRA

"INCLUDE" File Descriptions

In this program, INCLUDE files are used to ease the program development and maintenance. Any changes to an INCLUDE file are propagated throughout the program units that reference the file.

ELMSIZ.PRM Description

This file contains a single PARAMETER statement that sets the size of three constants. These parameterized constants in turn set the sizes of most of the arrays referenced in the program modules.

MXELEM specifies the maximum number of elements that will be allowed in the water column. This constant can be changed to tailor the program to a particular problem, by noting that each increase of one element will add approximately 400 (octal number) bytes to the program's memory requirements (assuming that the overlay descriptor is not altered).

The maximum number of bed layers is controlled by MAXLEV. Adjusting this parameter does not have much effect on the overall size of the program.

MAXCON limits the number of sediment and contaminant types that will be computed during a simulation and, while it is a parameter, it cannot be changed without program modifications. The program expects the following concentration values to be in the specified order:

1. sand
2. silt
3. clay
4. contaminant attached to sand
5. contaminant attached to silt
6. contaminant attached to clay
7. dissolved contaminant (not included in contaminant concentrations in river bed).

TRANS.COM Description

SERATRA minimizes the use of any COMMON areas of storage. However, because of the limits of the FORTRAN compiler, it was necessary to substitute the labeled COMMON TRANS for the argument list to the subroutine

TRANSP. Only those variables that are needed by the subroutine are included in the common. The sizes of the arrays are set by the PARAMETER statement in the file ELMSIZ.PRM.

DEFINITION OF VARIABLES

Variables used in SERATRA are listed here (name, array sizes, types, and definitions).

Name	Type	Definitions
ABAR(MXELEM)	R*4	Average surface area of the vertical elements (AREA(I)+AREA(I+1))/2.0
ALEN	R*4	River segment length
ALFA	R*4	Decay term
ANALMT	R*4	Cutoff concentration of dissolved contaminant for time series analysis
ANALYS	L*1	Control variable for time series analysis
AREA(MXELEM)	R*4	Surface area of each vertical element of a segment
AWID(MXELEM)	R*4	Width of each vertical element of a segment
B(MAXLEV,MAXCON-1)	R*4	Sediment and contaminant concentrations within the river bed
BASE(5)	BYTE	Result file base file name
BDIV	R*4	Standard thickness of each bed layer
BED	R*4	Total bed thickness
BETA	R*4	Source or sink term
C(MXELEM,MAXCON)	R*4	Sediment and contaminant concentrations in a water column
CCIN(MXELEM,MAXCON)	R*4	Sediment and contaminant concentrations flowing into a segment
CLAST(MXELEM,MAXCON)	R*4	Initial concentrations of sediment and contaminant for a given segment

Name	Type	Definitions
COLD(MXELEM,MAXCON)	R*4	Solution of the transport equations converted to cell-centered concentrations. They become inflow concentrations to the downstream segment.
CTRB(MXELEM,MACXON)	R*4	Vertically distributed sediment and contaminant concentrations in a tributary
CTRIB(MAXCON)	R*4	Cross-sectionally averaged concentrations of sediment and contaminant in a tributary
D(MXELEM)	R*4	Diagonal coefficients of the tridiagonal matrix
DECAY(6)	R*4	First-order degradation rates of contaminant
DELTD	R*4	Simulation time step in days
DELTH	R*4	Simulation time step in seconds
DELZ	R*4	Standard thickness of each vertical element
DELZT	R*4	Thickness of the top water element
DENS(3)	R*4	Sediment density
DEPO	R*4	Deposition rate of cohesive sediments
DEV(3)	BYTE	Device specification from the result file name
DFZ(4)	R*4	Vertical diffusion coefficients for sand, silt, clay and dissolved contaminant
DIAM(3)	R*4	Particle diameters of sand, silt and clay
DSHR(3)	R*4	Critical shear stress for deposition of sand, silt and clay
D1(MXELEM)	R*4	Subdiagonal coefficients of the tridiagonal matrix
D2(MXELEM)	R*4	Superdiagonal coefficients of the tridiagonal matrix
D50	R*4	Median bed sediment diameter
E(18)	R*4	Light adsorption coefficients for 18 different wavelengths

Name	Type	Definitions
ECHO	L*1	Line printer echo control variable
ELEV	R*4	Bed elevation of the segment
ENDHYD	I*4	Ending time (seconds) for the current set of hydrologic data
ENDIC	I*4	Ending time (seconds) for the current set of initial conditions
ENDTRB	I*4	Ending time (seconds) for the current set of tributary data
ERODE(3)	R*4	Erodibility coefficient of cohesive sediment
ETIME	I*4	Elapsed simulation time (seconds)
FERROR	L*1	Fatal error flag
FTYPE(3)	BYTE	File extension from the result file name
GSI	R*4	Total capacity of sediment transport rate
GUIC(3)	BYTE	Group number from the UIC of the result file name
HLDERR(100)	BYTE	Holding array for input error numbers
HRAD	R*4	Hydraulic radius
ILAYER(3)	I*2	Zero or positive value corresponds to the number of layers to be completely scoured for each sediment size fraction. Negative values correspond to the number of layers to be created by sediment deposition.
INFLO	I*2	Logical unit number of the file containing the results of the previous segment
ISEG	I*2	Segment number currently being simulated
ITPRT	I*2	Time plane save frequency
JULIAN	I*2	Julian starting date of the simulation
KAY1	R*4	Light extinction coefficient in clear water
KAY2	R*4	Light extinction coefficient due to suspended sediment in water

Name	Type	Definitions
MAXLEV	I*2	Maximum allowable number for bed layers
MAXCON	I*2	Total number of substances (sediment and contaminant) for simulation
MXELEM	I*2	Maximum allowable number of vertical water elements
NBED	I*2	Number of bed layers
NELEM	I*2	Number of elements in the water column for the current segment
NELEMB	I*2	Number of elements in the water column for the previous segment
NEWQI	L*1	QHIN data flag that is turned on when new sets of incoming flows or tributaries are supplied
NSETS	I*2	Number of time planes that the initial conditions of Segment 1 must be written to the input file of Segment 2. This is to allow for the travel time of the flow through the segment to reduce or eliminate the numerical dispersion.
NSTEPS	I*4	Total number of time steps for the simulation
NTRIBS	I*2	Indicator to signal the presence of a tributary to the segment
NUMERR	I*2	Input error counter
NXEQ	I*4	Current time step counter
OUTFLO	I*2	Logical unit number of the file receiving the results of each time step for the current segment
PCOEF(4)	R*4	Photolysis rate at the water surface for four seasons
PDELZ	R*4	Standard element thickness of the previous segment
PDELZT	R*4	Thickness of the top element of the previous segment

Name	Type	Definitions
PDEPTH	R*4	Flow depth of the previous segment
PELEV	R*4	Bed elevation of the previous segment
PHI	R*4	The reaction quantum yield for the contaminant in air-saturated pure water to be used to compute the photolysis rate
POR	R*4	Bed sediment porosity
PTDELZ	R*4	Standard element thickness at the previous time step
PTDLZT	R*4	Thickness of the top element at the previous time step
PTQAVG(MXELEM)	R*4	Discharge leaving the segment at the previous time step
QAVG(MXELEM)	R*4	Discharge leaving the segment at the end of the current time step
QHIN(MXELEM)	R*4	Inflow rate to the current segment
QHOLD(MXELEM)	R*4	Discharge into the segment from the upstream segment previously simulated
QHOUT(MXELEM)	R*4 .	Discharge leaving the segment at the end of the current time step
QV(MXELEM)	R*4	Vertical flow rate
R(MXELEM)	R*4	Finite-element load vector
RESELN	R*4	Water surface elevation
RHO	R*4	Density of water
RSUSP(3)	R*4	Erosion rates of sediments
S(MXELEM,3)	R*4	Finite-element unsymmetric band matrix
SCSHR(3)	R*4	Critical shear stresses for scour for sand, silt and clay
SD(MXELEM,3)	R*4	Amount of sediment removed from each element as a result of sediment deposition
SECDAY	R*4	Number of seconds in a day

Name	Type	Definitions
SECYR	I*4	Number of seconds in a year
SI(18,4)	R*4	Solar intensity table for 18 different wavelengths and the four seasons to be used to compute a photolysis rate
SIMLEN	I*4	Simulation length in seconds
SLOPE	R*4	River bottom slope
SMETH	BYTE	Sand transport method indicator
SORBK(9)	R*4	Distribution coefficients and transfer rate contaminants for adsorption mechanisms
SR(3)	R*4	Amount of sediment being added to each element due to bed sediment erosion
STRESS	R*4	Bed shear stress value
T	R*4	Ratio of the previous depth to the current depth
TEMPR	R*4	Water temperature
TRBOPT	I*2	Tributary input control variable
USTAR	R*4	Bed shear velocity
UUIC(3)	BYTE	User number from the UIC of the result file name
VEL	R*4	Flow velocity
VEL1	R*4	First convective term
VEL2	R*4	Second convective term
VOL	R*4	Segment flow volume
VSET(3)	R*4	Settling velocities of sediments
WL(18)	R*4	18 different sunlight wavelengths that are set into the photolysis submodel
XNT(3)	R*4	Bed sediment weight in a bed layer for each sediment size fraction
XYSO	R*4	Thickness of the top bed layer

FILE EXPLANATION

At any given time during a simulation, SERATRA could have as many as nine active files. These include the input stream data file, print file, temporary internal files, error message file and result files as shown in Table 1.

TABLE 1. LIST OF FILES

Name	LUN	Type	Usage
User Supplied	1	Formatted, sequential	Simulation input data file
DUMMY.DT1	2	Unformatted, sequential	Output of odd-numbered segments, input to even-numbered segments
DUMMY.DT2	3	Unformatted, sequential	Output of even-numbered segments, input to odd-numbered segments
HYDROLOGY.TMP	4	Unformatted, sequential	Temporary hydrologic data file
User Supplied	5	Unformatted, sequential	Simulation result file
SED.LST	6	Formatted, sequential	Print file
TRIBUTARY.TMP	7	Unformatted, sequential	Temporary tributary data file Interactive I/O logical unit
TIMSERIES.DAT	8	Unformatted, sequential	Time series analysis data
SERATERR.MSG	10	Formatted, direct access	Input error message file

The following describes each of these files.

Input Stream File

This file is prepared by the user prior to the simulation. The file contains all input data necessary to run the simulation. The file is opened by Subroutine STRTUP and is read by all of the input stream processing modules.

DUMMY.DT1 and DUMMY.DT2

These two files work in tandum to provide input to the current segment and to store the simulation results of each time step. Both files are opened in Subroutine STRTUP. When the initial conditions for Segment 1 are encountered in the input stream, they are written to DUMMY.DT1 by Subroutine UPSDAT. As Segment 1 is being simulated, its initial conditions are read from DUMMY.DT1 and after each time step its computed concentrations and outflows are written to DUMMY.DT2. The executive routine is responsible for writing the data to DUMMY.DT2 and Subroutine ICFLO reads the data from the inflow file.

When simulation of Segment 1 is completed and Segment 2 is to be modeled, the two files switch roles. DUMMY.DT2 becomes the inflow file and DUMMY.DT1 becomes the outflow file. This alternating pattern is then repeated for subsequent segments. DUMMY.DT1 is the inflow file for odd-numbered segments and the outflow file for even-numbered segments. DUMMY.DT2 is the outflow file for the odd segments and the inflow file for the even segments. The variables INFLO and OUTFLO control this swapping of reading and writing.

HYDROLOGY.TMP

This temporary file is used to store the hydrologic data for the segment being simulated. Subroutine STRTUP opens the file; Subroutine HYDDAT writes the data to the file from the input stream; and two modules read the data. SERATRA reads enough of the file to get the initial thickness of the top element and then rewinds the file so that it can be processed by Subroutine HYDFLO at the proper times.

Result Files

Each segment has a separate result file with a name based upon the base file name supplied by the user and the segment number. SERATRA opens the file and writes the first record that contains only the segment number. The rest of the data is processed and written to the files by Subroutine SAVEIT at predetermined time steps. These are the files that are processed by the post processing program SPPR.TSK.

SED.LST

This is the data file that receives all the data for printout, including the input stream echo and any error messages. The user must have this file printed.

TRIBUTARY.TMP

Any tributary data encountered in the input stream are written to this file by Subroutine TRBDAT. The data are read at each segment having a tributary during the simulation by Subroutine TRBFLO.

TIMESERIES.DAT

The executive routine performs all the processing of this file which will contain the data needed by the time analysis model FRANCO. It is recommended that the user rename this file after the simulation to avoid confusion later.

SERATERR.MSG

This direct access file contains all of the error messages that may be needed during the processing of the input stream. The file is used by Subroutine RPTERR and is created with Program MSGENT.TSK.

SECTION 3

DESCRIPTION OF MODEL OPERATION

SERATRA OPERATING INSTRUCTIONS

These operating instructions assume the operator is familiar with the RSX-11D operating system and can log onto the system and specify any nondefault devices or user identification codes. In the samples below, the underlined portions are the operator's response. SERATRA is activated by using the MCR RUN command.

MCR RUN SERATRA<alt>mode

After it has been loaded, SERATRA will seek information from the user concerning file names and simulation parameters.

ENTER NAME OF INPUT FILE > 3YEAR.CNL

What is needed here is the file descriptor of the input file that has been prepared by the user as described elsewhere in this report (SERATRA Input Requirements).

DO YOU WANT THE INPUT FILE ECHOED (Y OR N) > Y

As indicated, the proper response to this question is either a "Y" or an "N" This listing can consume a large number of pages if there is a large amount of time varying data. The listing does not go directly to the printer device but is written to the file "SED.LST" and the user must have the file printed.

ENTER BASE FILE NAME > 3YEAR.RLT

Simulation results are written at preselected time planes to files that are to be used during post-processing with Program SPPR. Each river segment being simulated is assigned a separate file to receive the results when the user supplies a portion of the file names (the base file name) and SERATRA generates the unique portion of the file name (the river segment number). The base file name consists of a 5-digit name (it must have exactly 5 digits) and a 3-digit extension such as "3YEAR.RLT". SERATRA appends a 4-digit segment number to each file it creates. For example, if three segments are being simulated, the files 3YEAR0001.RLT, 3YEAR0002.RLT and 3YEAR0003.RLT would be produced.

WHICH SAND CAPACITY METHOD IS TO BE USED?
ENTER T (TOFFALETTI) OR C (COLBY) > T

The user has the option of using either the Toffaletti or Colby method to compute the capacity of the flow to transport sand. When the Colby method is selected, it may fail under certain conditions during the simulation. If this happens, SERATRA will switch automatically to the Toffaletti method, but it will continue to try to use the Colby method whenever possible. The Colby method includes effects of water temperature and fine sediment concentration on sediment transport explicitly and is best suited to a small river. The Toffaletti method is applicable to both large and small rivers.

CONSOLE SWITCHES

Once each time step SERATRA examines the status of Console Switches 1 and 2 which are used to report simulation status and to terminate the run. The simulation status report consists of the line

SEGMENT #XXX TIME STEP YYYYYYYYYY

This line will be printed at the terminal device that was used to activate the model as long as Console Switch 1 is on. XXX is the number of the segment being simulated and YYYYYYYYYY is the number of the time step about to be taken.

Console Switch 2 is used to terminate the simulation at the operator's discretion. When the switch is raised the following message will be displayed.

***** SERATRA *****
TERMINATED BY OPERATOR AFTER TIME PLANE #XXXXXXXXXX IN SEGMENT #YYYYY

XXXXXXXXXX is the number of the time plane just completed and YYYYY is the number of the segment being simulated.

INPUT REQUIREMENTS FOR SERATRA

The input requirements for SERATRA consist of 18 separate data sets. Most of them are required and some are optional. Each data set is described separately in this section, but the order is implied by the number of the data set. In addition to numbers and titles of the data sets, each description contains a brief explanation of the data, the order and format of the records and the definition of each field in the records. Examples of each data set are also included. Actual values used in these examples must be regarded as only for illustration purposes.

Listed below are the numbers and titles of each data set. Data Sets 1, 2, 14 and 17 are read only once during the simulation, and the remaining sets must be repeated as a group for each river segment being simulated.

Data Set Numbers	Title
1	Simulation Identification Title
2	General Information Common to All Segments
3	Water Columns, Bed and Sediment Dimensions
4	Surface Area of Each Element
5	Particle Settling Velocity
6	Density of Sediment Particles
7	Diameter of Sediment Particles
8	Critical Shear Stresses for Bed Scouring
9	Critical Shear Stresses for Sediment Deposition
10	Erodibility Coefficients
11	Vertical Diffusion Coefficients
12	Adsorption/Desorption Values
13	Contaminant Degradation and Decay Parameters
14	Photolysis Coefficients and Tables
15	Initial Bed Conditions
16	Initial Water Column Conditions
17	Upstream Inflow Conditions to Segment 1
18	Tributary Inflow Conditions of Sediment and Contaminants
19	Hydrologic Data

The following table shows the order of each data set of an input file that would be required to model three river segments:

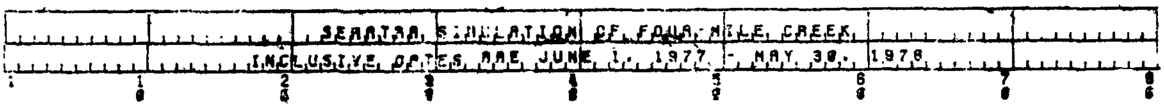
Sequence Numbers	Contents
1-2	Data Sets 1-2 (Data common to all segments)
3-13	Data Sets 3-13 (Data for Segment 1)
14	Data Set 14 (Common to all segments)
15-19	Data Sets 15-19 (Data for Segment 1)
20-29	Data Sets 3-13 (Data for Segment 2)
30-31	Data Sets 15-16 (Data for Segment 2)
32-33	Data Sets 18-19 (Data for Segment 2)
34-43	Data Sets 3-13 (Data for Segment 3)
44-45	Data Sets 15-16 (Data for Segment 3)
46-47	Data Sets 18-19 (Data for Segment 3)

Data Set No. 1 - Simulation Identification Title

This data set consists of two records that are used to identify both the simulation and the data file itself.

FORMAT: 20A4/20A4

Example:

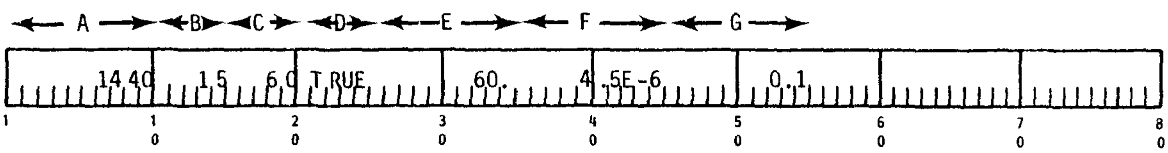


<u>Data Set No. 2 - General Information Common to All Segments</u>

FORMAT: I10,2I5,L5,3F10.0

Column	Variable	Description
1-10	NSTEPS	Number of time steps to be taken during the simulation (>0).
11-15	NSEG	Number of river segments to be simulated ($0<NSEG \leq 35$).
16-20	ITPRT	Time plane result save frequency. Because of the length of the simulations, it is often impractical to save sediment and contaminant concentrations after each time plane. This restricts the writing of the results to the file to every ITPRTth time plane. This value must be an even factor of NSTEPS or some of the simulation results will be lost ($0<ITPRT\leq NSTEPS$).
21-25	ANALYS	Time series analysis control variable =TRUE; Computed results at each time step satisfying, the cutoff limit of ANALMT will be written to a file that can be analyzed by Program FRANCO. =FALSE; None of the simulation results will be saved for future analysis.
26-35	DELTH	Time step length in SECONDS (DELTH>0.0).
36-45	ANALMT	Lower limit of depth-averaged dissolved concentrations used for the time series analysis. This variable sets a lower limit above which the results are saved for the time series analysis program, FRANCO. This result will only be saved when the dissolved concentration exceeds this limit. Care must be taken to assure that this value is not set too high. This variable is used only when ANALYS=TRUE. (kg/m^3 or pC/m^3)
46-55	DEPMIN	Minimum depth below which certain calculations will not be performed.

Example:



The above example describes that:

a) 1440 time steps will be taken for each of the
b) 15 segments; the results will be saved every
c) 60 time steps;

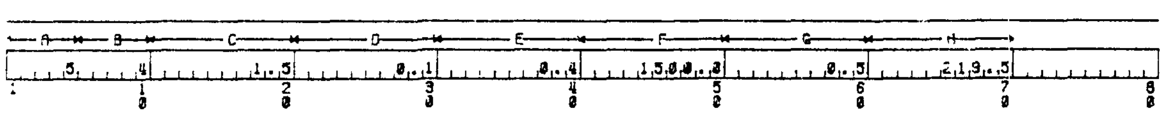
d) the results are to be made available to the time series analysis program, FRANCO;
e) 60 seconds is the length of each time step; and data will be saved for FRANCO when the dissolved concentration exceeds
f) 4.5E-6 (kg/m^3 or pC/m^3)
g) 0.1 meter is the lower limit for depth related calculations

Data Set No. 3 - Water Column, Bed and Segment Dimensions

FORMAT: 2I5,7F10.0

Column	Variable	Description
1-5	NELEM	Number of vertical water column elements (cells) in the segment at the beginning of the simulation. The maximum allowable value of this variable is set by the PARAMETER MXELEM which is kept in File ELMSIZ.PRM.
6-10	NBED	Number of bed layers at the beginning of the simulation ($0 < NBED \leq 10$)
11-20	DELZ	Thickness of all but the top vertical elements in meters ($0 < DELZ$)
21-30	BDIV	Thickness of all but the top bed layer in meters ($0.0 < BDIV$)
31-40	BED	Initial total bed thickness including the top bed layer in meters ($0.0 < Bed < NBED*BDIV$). The thickness of the top bed layer (XYSO) will be calculated by: XYSO = BED $\leq$(NBED-1)*BDIV
41-50	ALEN	Length of the segment in meters.
51-60	ELEV	Bed elevation of the segment measured at its midpoint in meters.
61-70	POR	Porosity of the bed sediment defined as the ratio of the water volume to the total volume. ($POR \leq 1.0$)
71-80	PELEV	Bed elevation upstream of Segment 1 (uppermost segment). This variable is read only with the data for Segment 1 and is ignored for all the other segments. It also tells SERATRA which method to use when computing the bed shear stress values. =0.0; bed shear stress computations will be based upon the velocity, depth and bed sediment size. This method is applicable for modeling impoundment areas such as reservoirs. ≠0.0; bed shear stress computations will be based upon the properties of free-flowing water (bed slope, hydraulic radius and specific weight of the water) in streams and rivers

Example:



The above example indicates that:

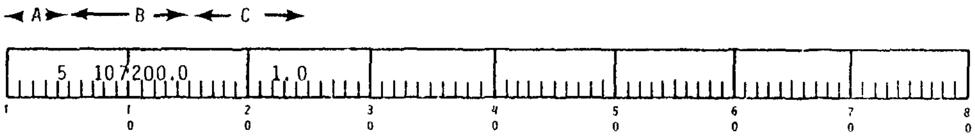
a) the segment initially contains 5 vertical elements;
b) the river bed initially consists of 4 layers;
c) the standard water column element thickness is 1.5 meters;
d) the standard thickness of the bed layers of 0.1 meter;
e) the total bed thickness is 0.4 meter which will leave a top bed layer thickness of 0.1 meter;
f) the segment's length is 1500.0 meters;
g) the porosity of the bed sediment is 0.5 or 50% empty space;
h) the bed elevation at the segment's midpoint is 219.5 meters (this also assigns the method to compute bed shear stresses).

Data Set No. 4 - Surface Areas of Each Element

Record 1 - Format I5,2F10.0

Column	Variable	Description
1-5	NAREA	Number of surface areas to be read in Record 2
6-15	VPAREA	Surface area to be used in excess of input data in square meters
16-25	DELEV	Vertical interval to be used in conjunction with VPAREA in meters

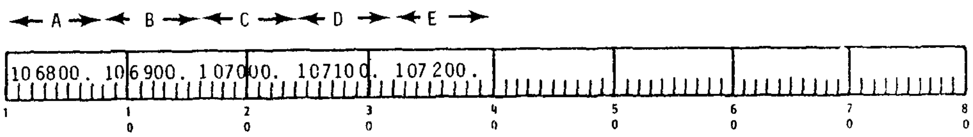
Example:



Record 2 - Format 8F10.0

Column	Variable	Description
1-10	AREA(1)	Surface area of channel bottom in square meters
11-20	AREA(2)	Surface area at node 2 in square meters
.		
.		
.		
.		
71-80	AREA(8)	Surface area at node 8 in square meters
.		
.		
.		
.		
	AREA (NAREA)	Surface area at top node in square meters

Example:

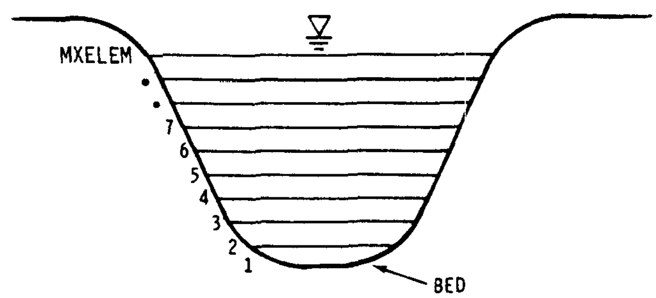


Supply surface areas AREA (1) through AREA (NAREA). Element numbers start from the river bottom. A value of MXELEM is specified in Include Statement ELMSIZ.RPM. Although value of MXELEM is adjustable, currently it is set to be 21.

Example explanation:

The surface areas of the first 8 elements are:

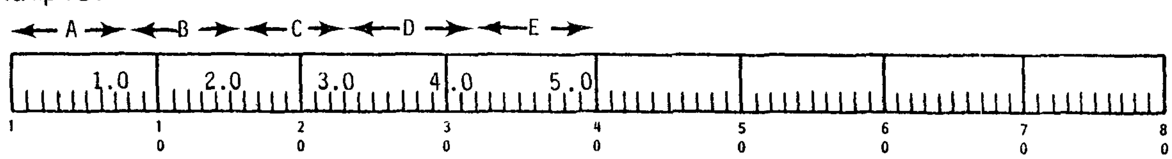
Element	Area (meters)
1	106800.0
2	106900.0
3	107000.0
4	107100.0
5	107200.0
6	107300.0
7	107400.0
8	107500.0



Record 3 - Format 8F10.0

Column	Variable	Description
1-10	EL(1)	Elevation or depth at channel bottom in meters
11-20	EL(2)	Elevation or depth at node 2 in meters
.		
.		
.		
.		
71-80	EL(8)	Elevation or depth at node 8 in meters
.		
.		
.		
.		
	EL(NAREA)	Elevation or depth at top node in meters

Example:



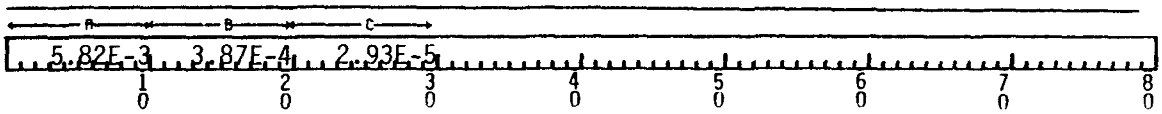
Data Set No. 5 - Particle Settling Velocity

FORMAT: 3F10.0

Column	Variable	Description
1-10	VSET(1)	Settling velocity of sand particles in m/sec; VSET(1) > 0.0) .

11-20	VSET(2)	Settling velocity of silt particles in m/sec; VSET(2) > 0.0) .
21-30	VSET(3)	Settling velocity of clay particles in m/sec; VSET(3) > 0.0) .

Example:



Example explanation:

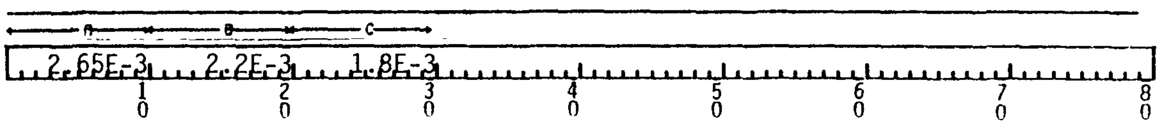
a) Settling velocity of sand is 5.82E-3 m/sec.
b) Settling velocity of silt is 3.87E-4 m/sec.
c) Settling velocity of clay is 2.93E-5 m/sec.

Data Set No. 6 - Density of Sediment Particles

FORMAT: 3F10.0

Column	Variable	Description
1-10	DENS(1)	Density of sand particles in kg/m^3; DENS(1) 0.0 .
11-20	DENS(2)	Density of silt particles in kg/m^3; DENS(2) 0.0 .
21-30	DENS(3)	Density of clay particles in kg/m^3; DENS(3) 0.0 .

Example:



Example explanation:

a) Density of sand is 2.65E+3 kg/m^3.
b) Density of silt is 2.2E+3 kg/m^3.
c) Density of clay is 1.8E+3 kg/m^3.

Data Set No. 7 - Diameter of Sediment Particles

FORMAT: 4F10.0

Column	Variable	Description
1-10	DIAM(1)	Diameter of sand particles in meters DIAM(1) 0.0 .
11-20	DIAM(2)	Diameter of silt particles in meters DIAM(2) 0.0 .
21-30	DIAM(3)	Diameter of clay particles in meters DIAM(3) 0.0 .
31-40	D50	Median bed sediment diameter in meters (D50 0.0).

Example:

```
<----A----><----B----><----C----><----D---->
  1.25E-4   3.00E-5   2.00E-6   1.25E-4
         1         2         3         4         5         6         7         8
         0         0         0         0         0         0         0         0
```

Example explanation:

a) Diameter of sand is 1.25E-4 meters.
b) Diameter of silt is 3.00E-5 meters.
c) Diameter of clay is 2.00E-6 meters.
d) Median bed sediment diameter is 1.25E-4 meters.

Data Set No. 8 - Critical Shear Stresses for Sediment Scouring

FORMAT: 3F10.0

Column	Variable	Description
1-10	SCSHR(1)	Critical shear stress for scouring sand in kg/m^2. This value will be disregarded in the model.
11-20	SCSHR(2)	Critical shear stress for scouring silt in kg/m^2.
21-30	SCSHR(3)	Critical shear stress for scouring clay in kg/m^2.

Example:

```
<----A----><----B----><----C---->
  1.00E-5   7.14E-3   7.45E-3
         1         2         3         4         5         6         7         8
         0         0         0         0         0         0         0         0
```

Example Explanation:

a) Critical shear stress value for scouring sand is 1.00E-5 kg/m^2.
b) Critical shear stress value for scouring silt is 7.14E-3 kg/m^2.
c) Critical shear stress value for scouring clay is 7.45E-3 kg/m^2.

Data Set No. 9 - Critical Shear Stresses for Sediment Deposition

FORMAT: 3F10.0

Column	Variable	Description
1-10	DSHR(1)	Critical shear stress value for depositing sand in kg/m^2 [DSHR(1) < SCSHR(1)]. This value will not be used in the model.
11-20	DSHR(2)	Critical shear stress value for depositing silt in kg/m^2 [DSHR(2) < SCSHR(2)].
21-30	DSHR(3)	Critical shear stress value for depositing clay in kg/m^2 [DSHR(3) < SCSHR(3)].

Example:

```
<----A---><----B---><----C--->
   1.00E-6   5.10E-4   2.04E-4
         1         2         3         4         5         6         7         8
         0         0         0         0         0         0         0         0
```

Example Explanation:

a) Critical shear stress value for depositing sand is 1.00E-6 kg/m^2.
b) Critical shear stress value for depositing silt is 5.10E-4 kg/m^2.
c) Critical shear stress value for depositing clay is 2.04E-4 kg/m^2.

Data Set No. 10 - Erodibility Coefficients

Format: 3F10.0

Column	Variable	Description
1-10	ERODE(1)	Erodibility coefficient of sand in kg/m^2-sec [ERODE(1) > 0.0]. This value will be disregarded in the model.
11-20	ERODE(2)	Erodibility coefficient of silt in kg/m^2-sec [ERODE(2) > 0.0].
21-30	ERODE(3)	Erodibility coefficient of clay in kg/m^2-sec [ERODE(3) > 0.0].

Example:

```
<----A---><----B---><----C--->
    1.5E-1    1.5E-1    1.5E-1
         1         2         3         4         5         6         7         8
         0         0         0         0         0         0         0         0
```

Example Explanation:

a) Sand erodibility coefficient is 1.5E-1 kg/m^2-sec.
b) Silt erodibility coefficient is 1.5E-1 kg/m^2-sec.
c) Clay erodibility coefficient is 1.5E-1 kg/m^2-sec.

Data Set No. 11 - Vertical Diffusion Coefficients

Format: 4F10.0

Column	Variable	Description
1-10	DFZ(1)	Vertical diffusion coefficient for sand in m^2/sec.
11-20	DFZ(2)	Vertical diffusion coefficient for silt in m^2/sec.
21-30	DFZ(3)	Vertical diffusion coefficient for clay in m^2/sec.
31-40	DFZ(4)	Vertical diffusion coefficient for dissolved contaminant in m^2/sec.

Example:

A	B	C	D
3.0E-4	3.0E-4	3.0E-4	3.0E-4

Example explanation:

a) Vertical diffusion coefficient for sand is 3.0E-4 m^2/sec.
b) Vertical diffusion coefficient for silt is 3.0E-4 m^2/sec.
c) Vertical diffusion coefficient for clay is 3.0E-4 m^2/sec.
d) Vertical diffusion coefficient for dissolved contaminant is 3.0E-4 m^2/sec.

Note: DFZ DELZ $\star$ WS

Data Set No. 12 - Adsorption Values (two records)

Record No. 1 - FORMAT: 8F10.0

Column	Variable	Description
1-10	SORBK(1)	Distribution coefficient of contaminant associated with sand in m^3/kg.
11-20	SORBK(2)	Distribution coefficient of contaminant associated with silt in m^3/kg.
21-30	SORBK(3)	Distribution coefficient of contaminant associated with clay in m^3/kg.
31-40	SORBK(4)	Contaminant transfer rate with suspended sand in sec^{-1}.
41-50	SORBK(5)	Contaminant transfer rate with suspended silt in sec^{-1}.
51-60	SORBK(6)	Contaminant transfer rate with suspended clay in sec^{-1}.
61-70	SORBK(7)	Contaminant transfer rate with bed sand in sec^{-1}.
71-80	SORBK(8)	Contaminant transfer rate with bed silt in sec^{-1}.

Record No. 2 - FORMAT: F10.0

Column	Variable	Description
1-10	SORBK(9)	Contaminant transfer rate with bed clay in sec^{-1}

Example:

A	B	C	D	E	F	G	H
1.0	5.0E-3	1.5E-4	4.63E-5	4.63E-5	4.63E-5	3.1709E-8	3.1709E-8

I
3.1709E-8

Example Explanation:

a) Contaminant distribution coefficient (Kd) value of sand is 1.0 m^3/kg.
b) Contaminant distribution coefficient (Kd) value of silt is 5.0E+3 m^3/kg.
c) Contaminant distribution coefficient (Kd) value of clay is 1.5E+4 m^3/kg.
d) Contaminant transfer rate with suspended sand is 4.63E-3 sec^{-1}.
e) Contaminant transfer rate with suspended silt is 4.63E-3 sec^{-1}.
f) Contaminant transfer rate with suspended clay is 4.63E-3 sec^{-1}.
g) Contaminant transfer rate with bed sand is 3.1709E-8 sec^{-1}.
h) Contaminant transfer rate with bed silt is 3.1709E-8 sec^{-1}.
i) Contaminant transfer rate with bed clay is 3.1709E-8 sec^{-1}.

Desorption rates are input in the same format following the adsorption rates in the previous data record.

Data Set 13 - Contaminant Degradation and Decay Parameters

This data set consists of one or two records depending upon the value input as DECAY(2). All of the decay parameters except DECAY(1) can be summed into this parameter, DECAY(2), and then the others do not have to be input.

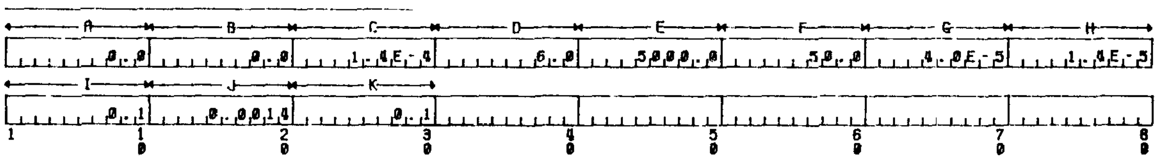
Record 1 - FORMAT: 8F10.0

Column	Variable	Description
1-10	DECAY(1)	Radionuclide decay (sec^{-1}).
11-20	DECAY(2)	Total degradation (sec^{-1}). ≠0.0; implies that all of the degradation parameters (except radionuclide decay) have been lumped into this rate and that the other degradation parameters will not be used. The second record will not be read. =0.0; implies that each of the degradation parameters will be input separately.
21-30	DECAY(6)	Volatilization degradation rate (sec^{-1}).
31-40	PH	Degree of acidity or alkalinity.
41-50	AKA	Second order acid rate constant for hydrolysis (sec^{-1}).
51-60	AKB	Second order base rate constant for hydrolysis (sec^{-1}).
61-70	AKN	Second order rate constant of neutral reaction with water (sec^{-1}).
71-80	AKOX	Second order rate constant of free radical oxygen for oxidation (M/sec).

Record 2 - FORMAT: 3F10.0; read only if DECAY(2) = 0.0;

Column	Variable	Description
1-10	RO2	Concentration of free radical oxygen (M).
11-20	AKBIO	Second order rate constant for biodegradation (m^3/kg/sec).
21-30	BIOMAS	Biomass per unit volume (kg/m^3).

Example:



A) Radionuclide decay is 0 sec^{-1}.
B) Total degradation is assigned to be 0.0 implying each of the degradation parameters to be followed.
C) Degradation rate due to volatilization is 1.4 x 10^{-4} sec^{-1}.
D) pH is 6.0.
E) Second order acid rate constant for hydrolysis is 5000 sec^{-1}.
F) Second order base rate constant for hydrolysis is 50 sec^{-1}.
G) Second order rate constant of neutral reaction with water is 4.0 x 10^{-5} sec^{-1}.
H) Second order rate constant of free radical oxygen for oxidation is 1.4 x 10^{-5} M/sec.
I) Concentration of the radical oxygen is 0.1 M.
J) Second order rate constant of free radical oxygen for oxidation is 0.0014, m^3/kg-sec.
K) Biomass per unit volume is 0.1 kg/m^3.

Data Set 14 - Photolysis Coefficients and Tables

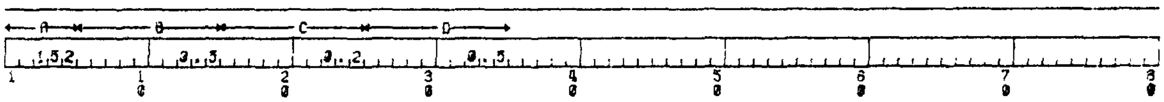
This data set is used to input the information necessary to compute chemical degradation due to photolysis. It consists of a record of coefficients, a table of adsorption coefficients and a table of solar intensities.

Record 1 - FORMAT: I5,3F10.0

Column	Variable	Description
1-5	JULIAN	Julian starting date of the simulation. The photolysis calculations can be turned off by inputting this variable as zero.
6-15	PHI	The reaction quantum yield for the chemical in air-saturated pure water. This is a measure of the efficiency with which a photochemical process converts adsorbed light into chemical reaction.

16-25	KAY1	Light extinction coefficient of clear water (m^{-1})
26-35	KAY2	Light extinction coefficient of suspended sediments in water (m^4/kg^2)

Example:



A) Julian starting data of the simulation is 152 days (June 1).
B) Reaction quantum yield of photolysis is 0.5.
C) Light extinction coefficient of clear water is 0.1 1/m.
D) Light extinction coefficient of suspended sediment in water is 0.5 1/m.

Records 2-4 - Adsorption Coefficient Table

These three records are the adsorption coefficients table for the 18 wavelengths. Each value is a measure of the chemical's ability to adsorb light of a particular wavelength. Wavelength units are nano meters.

Record 2 - FORMAT: 8F10.0

Column	Variable	Description
1-10	E(1)	Adsorption coefficient for the wavelength of 300.00
11-20	E(2)	Adsorption coefficient for the wavelength of 303.75
21-30	E(3)	Adsorption coefficient for the wavelength of 308.75
31-40	E(4)	Adsorption coefficient for the wavelength of 313.75
41-50	E(5)	Adsorption coefficient for the wavelength of 318.75
51-60	E(6)	Adsorption coefficient for the wavelength of 323.10
61-70	E(7)	Adsorption coefficient for the wavelength of 346.00
71-80	E(8)	Adsorption coefficient for the wavelength of 370.00

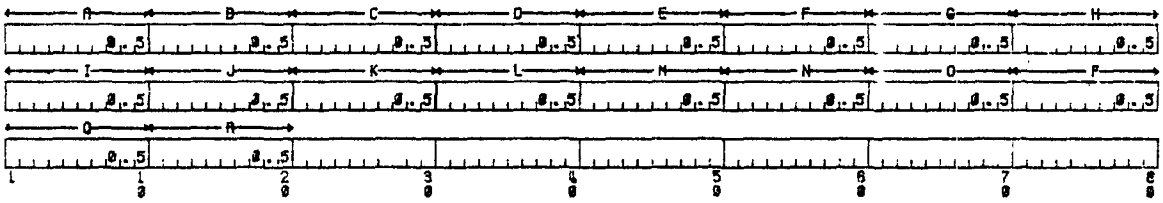
Record 3 - FORMAT: 8F10.0

Column	Variable	Description
1-10	E(9)	Adsorption coefficient for the wavelength of 400.00
11-20	E(10)	Adsorption coefficient for the wavelength of 430.00
21-30	E(11)	Adsorption coefficient for the wavelength of 460.00
31-40	E(12)	Adsorption coefficient for the wavelength of 490.00
41-50	E(13)	Adsorption coefficient for the wavelength of 536.25
51-60	E(14)	Adsorption coefficient for the wavelength of 587.50
61-70	E(15)	Adsorption coefficient for the wavelength of 637.50
71-80	E(16)	Adsorption coefficient for the wavelength of 687.50

Record 4 - FORMAT: 2F8.0

Column	Variable	Description
1-10	E(17)	Adsorption coefficient for the wavelength of 756.00
11-20	E(18)	Adsorption coefficient for the wavelength of 800.00

Example:



A) Molar absorption coefficient for the light with the wavelength of 300.00 is 0.5 1/M/cm.
B) Molar absorption coefficient for the light with the wavelength of 303.75 is 0.5 1/M/cm.
C) Molar absorption coefficient for the light with the wavelength of 308.75 is 0.5 1/M/cm.
D) Molar absorption coefficient for the light with the wavelength of 313.75 is 0.5 1/M/cm.
E) Molar absorption coefficient for the light with the wavelength of 318.75 is 0.5 1/M/cm.
F) Molar absorption coefficient for the light with the wavelength of 323.10 is 0.5 1/M/cm.
G) Molar absorption coefficient for the light with the wavelength of 346.00 is 0.5 1/M/cm.
H) Molar absorption coefficient for the light with the wavelength of 370.00 is 0.5 1/M/cm.
I) Molar absorption coefficient for the light with the wavelength of 400.00 is 0.5 1/M/cm.
J) Molar absorption coefficient for the light with the wavelength of 430.00 is 0.5 1/M/cm.
K) Molar absorption coefficient for the light with the wavelength of 460.00 is 0.5 1/M/cm.
L) Molar absorption coefficient for the light with the wavelength of 490.00 is 0.5 1/M/cm.
M) Molar absorption coefficient for the light with the wavelength of 536.25 is 0.5 1/M/cm.
N) Molar absorption coefficient for the light with the wavelength of 587.50 is 0.5 1/M/cm.
O) Molar absorption coefficient for the light with the wavelength of 637.50 is 0.5 1/M/cm.
P) Molar absorption coefficient for the light with the wavelength of 687.50 is 0.5 1/M/cm.
Q) Molar absorption coefficient for the light with the wavelength of 758.00 is 0.5 1/M/cm.
R) Molar absorption coefficient for the light with the wavelength of 800.00 is 0.5 1/M/cm.

Records 5 to 16 - Solar Intensity Table

This table consists of four sets of 18 wavelength values as shown below:

Record No.	Season	Calendar Dates	Julian Dates
5-7	Spring	Mar. 1-May 31	60-151
8-10	Summer	June 1-Aug. 31	152-243
11-13	Fall	Sept.1-Nov. 30	244-334
14-16	Winter	Dec. 1-Feb. 28	335-365; 1-59

The 18 values correspond to the wavelengths specified in the adsorption coefficient table (Records 2-4). The following tables provide suggested intensity values for 5 different latitudes that should cover most applications within the United States. These values are the result of research conducted by the Stanford Research Institute for the U.S. Environmental Protection Agency (Smith et al., 1977; Stanford Research Institute, 1979).

TABLE 2. SOLAR INTENSITY VALUES FOR LATITUDE 10 N

Wavelength,	Solar Intensity, einsteins/liter/day			
Nanometers	Spring	Summer	Fall	Winter
300	1.02E-2	4.66E-4	4.19E-4	3.20E-4
303.75	1.78E-2	3.16E-3	2.87E-3	2.39E-3
308.75	2.85E-2	9.37E-3	8.51E-3	7.26E-3
313.75	3.27E-2	1.90E-2	1.73E-3	1.51E-2
318.75	4.18E-2	2.91E-2	2.66E-2	2.38E-2
323.1	3.70E-2	2.65E-2	2.91E-2	2.36E-2
346	3.39E-1	3.29E-1	2.99E-1	2.92E-1
370	4.33E-1	4.38E-1	3.85E-1	3.44E-1
400	8.40E-1	8.37E-1	7.64E-1	6.96E-1
430	1.16	1.17	1.07	9.80E-1
460	1.47	1.47	1.36	1.23
490	1.50	1.50	1.37	1.27
536.25	2.74	2.69	2.46	2.26
587.5	2.90	2.79	2.52	2.35
637.5	2.90	2.80	2.60	2.43
687.5	2.80	2.80	2.60	2.30
756	2.70	2.70	2.50	2.40
800	3.00	2.50	2.30	2.10

TABLE 3. SOLAR INTENSITY VALUES FOR LATITUDE 20°N

Wavelength, Nanometers	Solar Intensity, einsteins/liter/day			
	Spring	Summer	Fall	Winter
300	3.51E-4	4.44E-4	2.74E-4	1.47E-4
303.75	2.51E-3	3.15E-3	2.20E-3	1.47E-3
308.75	8.09E-3	9.61E-3	6.89E-3	5.34E-3
313.75	1.81E-2	1.97E-2	1.48E-2	1.15E-2
318.75	2.82E-2	3.02E-2	2.33E-2	1.88E-2
323.1	2.83E-2	3.03E-2	2.33E-2	1.88E-2
340	3.29E-1	3.47E-1	2.68E-1	2.21E-1
370	4.24E-1	4.47E-1	3.45E-1	2.86E-1
406	8.41E-1	8.83E-1	6.96E-1	5.97E-1
430	1.17	1.23	9.80E-1	8.40E-1
460	1.47	1.55	1.24	1.06
490	1.50	1.58	1.26	1.09
536.25	2.68	2.81	2.30	1.95
587.5	2.80	2.96	2.35	2.03
637.5	2.80	2.90	2.42	2.07
687.5	2.80	3.00	2.40	2.10
750	2.76	2.80	2.20	2.36
800	2.50	2.70	2.26	1.60

TABLE 4. SOLAR INTENSITY VALUES FOR LATITUDE 30°N

Wavelength, Nanometers	Solar Intensity, einsteins/liter/day			
	Spring	Summer	Fall	Winter
300	2.30E-4	3.65E-4	1.35E-4	4.10E-5
303.75	2.13E-3	2.32E-3	1.44E-3	6.50E-4
308.73	7.26E-3	9.02E-3	4.84E-3	2.76E-3
313.75	1.65E-2	1.92E-2	1.16E-2	7.55E-3
318.75	2.64E-2	3.02E-2	1.89E-2	1.31E-2
323.1	2.69E-2	3.04E-2	2.30E-2	1.34E-2
340	3.20E-1	3.74E-1	2.23E-1	1.70E-1
370	4.14E-1	4.37E-1	2.84E-1	2.19E-1
400	8.27E-1	9.07E-1	6.23E-1	4.75E-1
430	1.15	1.34	8.50E-1	6.69E-1
460	1.45	1.59	1.09	8.50E-1
490	1.48	1.62	1.11	8.80E-1
636.25	2.64	2.89	2.00	1.57
587.5	2.74	3.03	2.07	1.63
637.5	2.76	3.00	2.09	1.67
687.5	2.80	3.00	2.10	1.73
750	2.70	2.90	2.10	1.63
800	2.50	2.80	1.90	1.60

TABLE 5. SOLAR INTENSITY VALUES FOR LATITUDE 40° N

Wavelength, Nanometers	Solar Intensity, einsteins/liter/day			
	Spring	Summer	Fall	Winter
300	1.09E-4	2.49E-4	1.09E-4	5.38E-6
303.75	1.37E-3	2.32E-3	1.37E-3	1.56E-4
308.75	2.96E-3	7.93E-3	5.35E-3	1.02E-3
313.75	7.99E-3	1.81E-2	1.38E-2	3.79E-3
318.75	1.38E-2	2.91E-2	2.319E-2	7.53E-3
323.1	1.42E-2	2.97E-2	2.39E-2	8.10E-3
340	1.78E-1	3.54E-1	1.08E-1	7.52E-2
370	2.30E-1	4.58E-1	3.84E-1	1.47E-1
400	5.26E-1	9.71E-1	7.91E-1	3.38E-1
430	6.76E-1	1.28	1.11	4.80E-1
460	8.90E-1	1.43	1.39	6.10E-1
490	9.23E-1	1.63	1.42	6.20E-1
536.25	1.69	2.92	2.52	1.12
587.5	1.73	3.05	2.62	1.16
637.5	1.78	3.00	2.60	1.19
687.5	1.50	3.10	4.70	1.39
750	1.70	2.90	2.60	1.20
800	1.60	2.90	2.50	1.16

TABLE 6. SOLAR INTENSITY VALUES FOR LATITUDE 50°N

Wavelength, Nanometers	Solar Intensity, einsteins/liter/day			
	Spring	Summer	Fall	Winter
300	3.71E-5	7.88E-6	1.52E-4	4.00E-7
303.75	7.10E-4	1.75E-3	2.25E-4	1.57E-5
308.75	3.55E-3	6.53E-3	1.29E-3	1.78E-4
313.75	7.30E-3	1.63E-2	4.39E-3	1.20E-3
318.75	1.84E-3	2.67E-2	8.64E-3	2.93E-3
323.1	1.96E-2	2.77E-2	9.20E-3	3.68E-3
340	2.66E-1	3.43E-1	1.24E-1	6.29E-2
370	3.48E-1	4.44E-1	1.66E-1	8.21E-2
400	7.24E-1	9.04E-1	3.65E-1	1.96E-1
430	1.02	1.26	5.17E-1	2.75E-1
460	1.29	1.60	6.60E-1	3.51E-1
470	1.32	1.63	6.80E-1	3.55E-1
536.25	2.34	2.90	1.22	6.30E-1
587.5	2.40	3.04	1.25	6.40E-1
637.5	2.44	3.00	1.31	6.90E-1
687.5	2.50	3.10	1.34	7.10E-1
750	2.50	2.90	1.31	7.10E-1
800	2.30	2.90	1.24	6.90E-1

Records 5, 8, 11, 14 - Format: 8F10.0

Column	Variable	Description
1-10	SI(1,I)	Solar intensity for wavelength of 300.00 (einsteins liter^{-1} day^{-1})
11-20	SI(2,I)	Solar intensity for wavelength of 303.75
21-30	SI(3,I)	Solar intensity for wavelength of 308.75
31-40	SI(4,I)	Solar intensity for wavelength of 313.75
41-50	SI(5,I)	Solar intensity for wavelength of 318.75
51-60	SI(6,I)	Solar intensity for wavelength of 323.10
61-70	SI(7,I)	Solar intensity for wavelength of 346.00
71-80	SI(8,I)	Solar intensity for wavelength of 370.00

Records 6, 9, 12, 15 - Format: 8F10.0

Column	Variable	Description
1-10	SI(9,I)	Solar intensity for wavelength of 400.00
11-20	SI(10,I)	Solar intensity for wavelength of 430.00
21-30	SI(11,I)	Solar intensity for wavelength of 460.00
31-40	SI(12,I)	Solar intensity for wavelength of 490.00
41-50	SI(13,I)	Solar intensity for wavelength of 536.25
51-60	SI(14,I)	Solar intensity for wavelength of 587.50
61-70	SI(15,I)	Solar intensity for wavelength of 637.50
71-80	SI(16,I)	Solar intensity for wavelength of 687.50

Records 7, 10, 13, 16 - Format: 2F10.0

Column	Variable	Description
1-10	SI(17,I)	Solar intensity for wavelength of 756.00
11-20	SI(18,I)	Solar intensity for wavelength of 800.00

Example:

A	B	C	D	E	F	G	H
2.49E-4	2.32E-3	7.93E-3	1.81E-2	2.91E-2	2.97E-2	3.54E-1	4.58E-1

I	J	K	L	M	N	O	P
3.71E-1	1.28	1.43	1.63	2.92	3.05	3.00	3.10

Q	R						
2.90	2.90						
10	20	30	40	50	60	70	80

This is the example of solar intensities during summer for latitude 40°N.

A) Solar intensity of the light with the wavelength of 300.00 is 2.49 x 10^{-4} einstein/liter/day.
B) Solar intensity of the light with the wavelength of 303.75 is 2.32 x 10^{-3} einstein/liter/day.

C) Solar intensity of the light with the wavelength of 308.75 is 7.93 x 10^{-3} einstein/liter/day.
D) Solar intensity of the light with the wavelength of 313.75 is 1.81 x 10^{-2} einstein/liter/day.
E) Solar intensity of the light with the wavelength of 318.75 is 2.91 x 10^{-2} einstein/liter/day.
F) Solar intensity of the light with the wavelength of 323.10 is 2.97 x 10^{-2} einstein/liter/day.
G) Solar intensity of the light with the wavelength of 346.00 is 3.54 x 10^{-4} einstein/liter/day.
H) Solar intensity of the light with the wavelength of 370.00 is 4.58 x 10^{-1} einstein/liter/day.
I) Solar intensity of the light with the wavelength of 400.00 is 9.71 x 10^{-1} einstein/liter/day.
J) Solar intensity of the light with the wavelength of 430.00 is 1.28 einstein/liter/day.
K) Solar intensity of the light with the wavelength of 460.00 is 1.43 einstein/liter/day.
L) Solar intensity of the light with the wavelength of 490.00 is 1.63 einstein/liter/day.
M) Solar intensity of the light with the wavelength of 536.25 is 2.92 einstein/liter/day.
N) Solar intensity of the light with the wavelength of 587.50 is 3.05 einstein/liter/day.
O) Solar intensity of the light with the wavelength of 637.50 is 3.00 einstein/liter/day.
P) Solar intensity of the light with the wavelength of 687.50 is 3.10 einstein/liter/day.
Q) Solar intensity of the light with the wavelength of 756.00 is 2.90 einstein/liter/day.
R) Solar intensity of the light with the wavelength of 800.00 is 2.90 einstein/liter/day.

Data Set 15 - Initial Bed Conditions

A set of initial conditions is read for each of the six parameters using one of two input methods. The first method is used when there is no vertical variation of the parameter. This is accomplished by inputting the control variable as a negative number which causes the specified value to be applied to each of the bed layers. When there is a vertical variation, the values for each layer must be specified. Bed layers are numbered upward from the bottom (Layer 1) to the bed surface layer (Layer BED). In the descriptions below, only the first parameter has been fully explained because the other five are input the same way.

Parameter 1 - Weight fraction of sand in river bed

Record 1 - Format: I5,F10.0

Column	Variable	Description
1-5	SWITCH	This is the control variable that signals the input method to be used. <0; there is no vertical variation and the weight fraction found in the variable VALUE is to be applied to each layer ≥0; the weight fraction varies with depth and the data in the record(s) that follow will contain the values to be applied to each bed layer.
6-15	VALUE	The weight fraction of sediment to be assigned to each vertical layer.

NOTE: The next two records are read only when SWITCH ≥ 0. NBED values will be read.

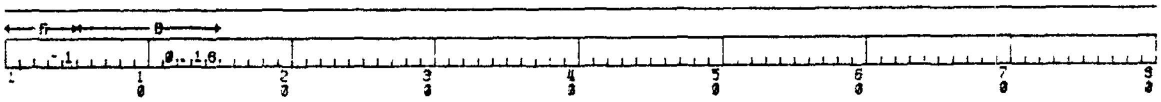
Record 2 - Format 8F10.0

Column	Variable	Description
1-10	B(1,1)	Weight fraction of sand in Layer 1 (bottommost element)
11-20	B(2,1)	Weight fraction of sand in Layer 2
21-30	B(3,1)	Weight fraction of sand in Layer 3
31-40	B(4,1)	Weight fraction of sand in Layer 4
41-50	B(5,1)	Weight fraction of sand in Layer 5
51-60	B(6,1)	Weight fraction of sand in Layer 6
61-70	B(7,1)	Weight fraction of sand in Layer 7
71-80	B(8,1)	Weight fraction of sand in Layer 8

Record 3 - Format 8F10.0; Read only if NBED > 8

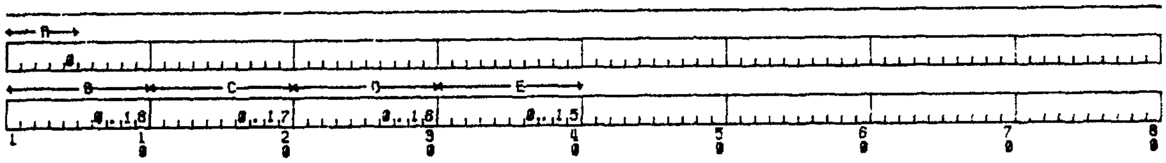
Column	Variable	Description
1-10	B(9,1)	Weight fraction of sand in Layer 9
11-20	B(10,1)	Weight fraction of sand in Layer 10

Example No. 1:



The above sample is of a case where there is
A) no vertical variation and the value of
B) 0.18 should be assigned to each bed layer.

Example No. 2:



This illustrates the second method when

- A) The sand fraction varies vertically. There are four vertical layers (NBED=4). These layers will then be assigned the value of
- B) 0.18 for Layer 1,
- C) 0.17 for Layer 2,
- D) 0.16 for Layer 3, and
- E) 0.15 for Layer 4. Note that Record 3 is not (and should not be) included.

The record order and format for the remaining five parameters are the same as that of the first and they are read in the following order:

Parameter 2 - Weight fraction of bed silt
Parameter 3 - Weight fraction of bed clay
Parameter 4 - Contaminant concentration associated with bed sand
Parameter 5 - Contaminant concentration associated with bed silt
Parameter 6 - Contaminant concentration associated with bed clay

The contaminant concentration units depend upon the type of contaminant:

Radionuclide - pC/kg
Pesticide and other toxic chemicals - kg/kg

<u>Data Set 16 - Initial Water Column Conditions</u>

This data set is used to preset the concentrations of sand, silt, clay and their associated contaminant concentrations at nodal points of each water column element. The user has the option of selecting one of two input methods depending upon the vertical distribution of each of the seven parameters. When the parameter does not vary vertically, it is possible to input only a constant that will be assigned to each element in the water column. Otherwise, the value for each element must be input element by element. The elements are numbered from the bottom (Element 2) to the water surface (Element NELEM). In the description below, only the first parameter is completely explained because the same form is used for the remaining six parameters.

Parameter 1 - Concentration of suspended sand (kg/m^3)

Record 1 - Format: I5,F10.0

Column	Variable	Description
1-5	SWITCH	This is the control variable that signals the input method to be used. <0; there is no vertical variation and the constant found in the variable VALUE is to be applied to each element in the water column. $\geq$0; the parameter varies with depth and the value for each element will be supplied.
6-15	VALUE	When SWITCH is less than zero; this is the constant concentration that will be given to each nodal point of the elements in the water column.

The concentration record described below must be repeated enough times to provide NELEM+1 values for the total number (NELEM) of elements. Of course, the concentration values are read only when SWITCH$\geq$0.

The contaminant concentration units depend upon the type of contaminant:

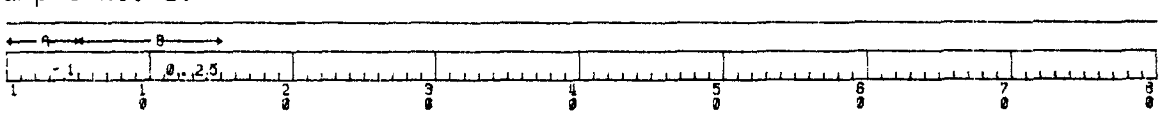
Radionuclide - $\frac{pC}{kg}$

Pesticide and other toxic chemicals - $\frac{kg}{kg}$

Concentration Record - Format 8F10.0

Column	Variable	Description
1-10	C(N,1)	Concentration of sand at the bottom nodal point of Element N
11-20	C(N+1,1)	Concentration of sand at the top nodal point of Element N and the bottom nodal point of Element N+1
21-30	.	.
.	.	.
.	.	.
71-80	C(N+7,1)	Concentration of sand at the bottom nodal point of Element N+7 (if Element N+7 exists) and the top nodal point of Element N+6

Example No. 1:

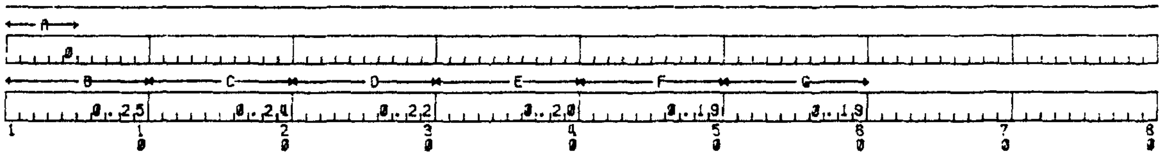


In the above case, there was

A) no vertical variation of the parameter and

B) 0.25 kg/m^3 will be assigned to each nodal point of element in the water column.

Example No. 2:



Above is a sample of:

A) Vertical variations of the parameter. The values to be assigned to each of the water elements are
B) 0.25 kg/m^3 at the bottom nodal point of Element 1 (bottommost element),
C) 0.24 kg/m^3 at the top nodal point of Element 1 and the bottom nodal point of Element 2
D) 0.22 kg/m^3 at the top nodal point of Element 2 and the bottom nodal point of Element 3
E) 0.20 kg/m^3 at the top nodal point of Element 3 and the bottom nodal point of Element 4
F) 0.19 kg/m^3 at the top nodal point of Element 4 and the bottom nodal point of Element/5 and
G) 0.19 kg/m^3 at the top nodal point of Element 5.

The record order and format described above are the same as those for the remaining six parameters which are read in the following order:

Parameter 2 - Concentration of silt (kg/m^3)
Parameter 3 - Concentration of clay (kg/m^3)
Parameter 4 - Concentration of contaminant associated with sand
Parameter 5 - Concentration of contaminant associated with silt
Parameter 6 - Concentration of contaminant associated with clay
Parameter 7 - Concentration of dissolved contaminant

The concentration units of Parameters 4, 5 and 6 depend upon the type of contaminant:

Radionuclide - pC/kg
Pesticide and other toxic chemicals - kg/kg

The dissolved concentration units are correspondingly pC/m^3 or kg/m^3.

Data Set 17 - Upstream Inflow Conditions to Segment 1

This data set supplies upstream boundary conditions of sediment and contaminant concentrations to Segment 1 (uppermost river segment). Since these data can vary with time, it is necessary to supply enough data to span the duration of the simulation (DELTH*NSTEPS). The user must remember that the data are used in the function of a step function. That is, the concentrations remain constant until the next is to be used. There is no interpolation between data sets.

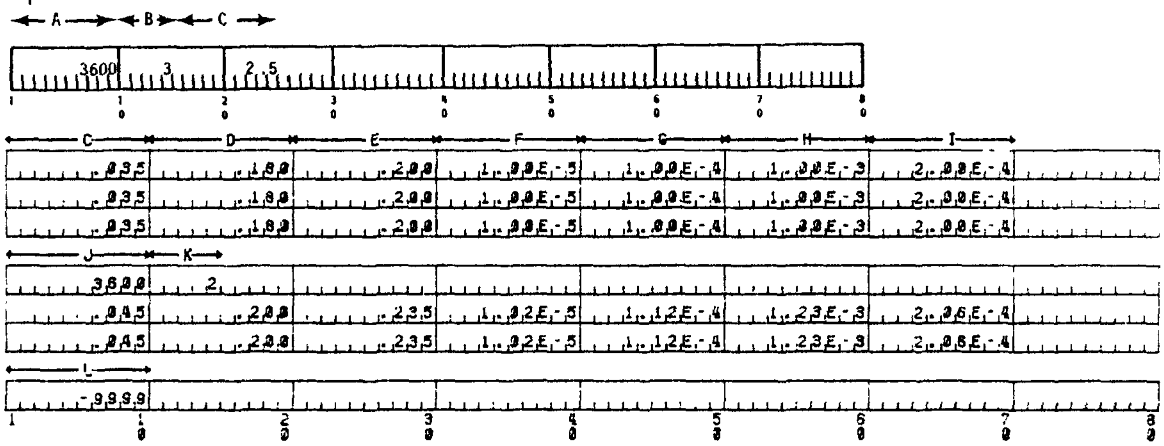
Record 1 - Format: I10, I5, F10.0

Column	Variable	Description
1-10	ENDTIM	Ending time for the data set that follows. This implies that the starting time for the first data set is T = 0 and the starting time of the second data set is ENDTIM+DELTH. A value of -9999 signals the end of the upstream inflow conditions (seconds).
11-15	NM	Number of vertical water column elements that data will be provided for. ≤0; there is no vertical variation and the constants found in the variable array CCIN are to be applied to each element in the water column >0; the parameters in the variable array CCIN vary with depth and the values for each element will be supplied
16-25	UDEPTH	Elevation or Depth upstream of study reach until T=ENDTIM in meters

Record 2 - Format 7F10.0; NM records will be used.

Column	Variable	Description
1-10	CCIN(I,1)	Concentration of sand for Element I (kg/m^3).
11-20	CCIN(I,2)	Concentration of silt for Element I (kg/m^3).
21-30	CCIN(I,3)	Concentration of clay for Element I (kg/m^3).
31-40	CCIN(I,4)	Concentration of contaminant associated with sand for Element I (pesticide and other chemicals - kg/kg or radionuclide - pC/kg).
41-50	CCIN(I,5)	Concentration of contaminant associated with silt for Element I (pesticide and other chemicals - kg/kg or radionuclide - pC/kg).
51-60	CCIN(I,6)	Concentration of contaminant associated with clay for Element I (pesticide and other chemicals - kg/kg or radionuclide - pC/kg).
61-70	CCIN(I,7)	Concentration of dissolved contaminant (pesticide and other chemicals - kg/m^3 or radionuclide - pC/m^3).

Example:



Two sets of upstream inflow conditions are shown in the above sample. The values of the first set will be used from the beginning of the simulation until one time step beyond.

A) 3600 seconds at which time the second set will be used. During this period the water column will consist of
B) 3 vertical elements. The values for each of the concentrations for the first (bottommost) element are as follows:
C) sand concentration of 0.035 kg/m^3
D) Silt concentration of 0.180 kg/m^3
E) clay concentration of 0.200 kg/m^3
F) particulate contaminant concentration associated with sand to be 1.0E-5 kg/kg
G) particulate contaminant concentration associated with silt to be 1.0E-4 kg/kg
H) particulate contaminant concentration associated with clay to be 1.0E-3 kg/kg
I) dissolved contaminant concentration of 2.0E-4 kg/m^3

The second sample indicates that the last set of data is to be used until the simulated time of

J) 36,000 seconds. Of course the length of the simulation may not exceed the 10 hour. During this period the river segment will have
K) two vertical elements. The data set termination signal is
L) -9999.

Record 3 - Format I5,2F10.0

Column	Variable	Description
1-5	NWID	Number of upstream widths to be read. If set to zero, the constant width, UWIDTH, will be used.
6-15	UWIDTH	Width to be used in excess of input data in meters
16-25	DEL	Vertical interval to be used in conjunction with UWIDTH

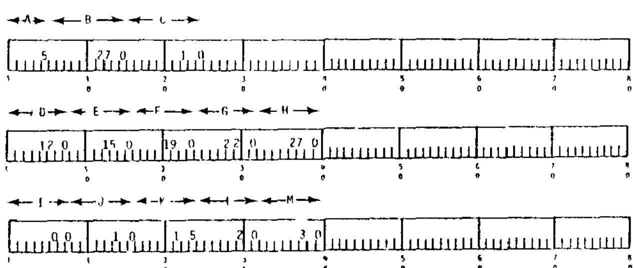
Record 4 - Format 8F10.0

Column	Variable	Description
1-10	UWID(1)	Width of channel bottom upstream of study reach in meters
11-20	UWID(2)	Width at node 2 upstream of study reach in meters
.		
.		
.		
.		
71-80	UWID(8)	Width at node 8 upstream of study reach in meters
.		
.		
.		
.		
	UWID(NWID)	Width at top node upstream of study reach in meters

Record 5 - Format 8F10.0

Column	Variable	Description
1-10	VEL(1)	Elevation or depth at channel bottom in meters
11-20	VEL(2)	Elevation or depth at node 2 in meters
.		
.		
.		
.		
71-80	VEL(8)	Elevation or depth at node 8 in meters
.		
.		
.		
.		
	VEL(NWID)	Elevation or depth at top node in meters

Example:



A) 5 sets of widths and elevations are to describe the channel geometry upstream of the study reach
B) if the computations require information at a depth larger than the data provided, a 27.0 meter width will be used
C) the width in B) will be used in elements having a vertical thickness of 1.0 meter
D)-H) upstream widths

I)-M)
elevations associated with the upstream widths

Data Set 18 - Tributary Influx Loads of Sediment and Contaminants

This data set is used to input sediment and contaminant contributions from a tributary. Provisions are provided for inputting time varying influx loads and the user must supply enough data to span the length of the simulation (DELTH*NSTEPS). During the simulation the concentrations are supplied as a step function. The units of contaminant contributions from the tributary are kg/sec for pesticides and other chemicals and pC/sec for radionuclides.

Record 1 - Format: 2I5

Column	Variable	Description
1-5	NTRIBS	Tributary indicator: =0; No tributary joining at this segment. =1; there is a tributary joining at this segment and its data follow.
6-10	TRBOPT	Tributary input control variable (ignored when NTRIBS = 0). =0; the user will supply one value for each sediment and contaminant load and the model will distribute it vertically uniformly. =1; the user will provide NELEM (see Data Set 3) values for each sediment or contaminant load to NELEM vertical elements.

The following two records make up a set that is used only when NTRIBS = 1. If input data are not time varying, only one set of data is necessary. The only requirement for this steady input case is that the ending time be equal to or greater than length of the simulation.

Record 2 - Format I10

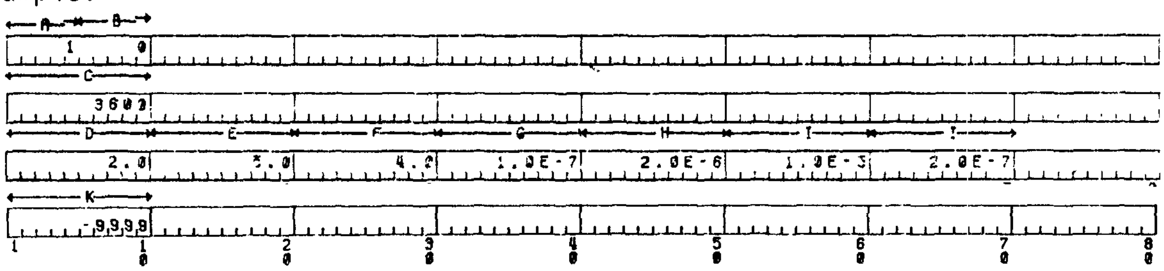
Column	Variable	Description
1-10	ENDTIME	Ending time (in seconds) of the data that follow. Influx loads described in the next record or records will be used until this time is exceeded. This variable is also used to signal the end of the tributary data. This is accomplished by inputting ENDTIM as "-9999".

Record 3 - Format 7F10.0

When TRBOPT = 0 only one record is expected and the model will redistribute the influx load vertically. If TRBOPT = 1, a record for each of the NELEM vertical elements must be supplied.

Column	Variable	Description
1-10	CTRB(J,1)	Influx load of sand (kg/sec)
11-20	CTRB(J,2)	Influx load of silt (kg/sec)
21-30	CTRB(J,3)	Influx load of clay (kg/sec)
31-40	CTRB(J,4)	Influx load of contaminant associated with sand (pC/sec or kg/sec)
41-50	CTRB(J,5)	Influx load of contaminant associated with silt (pC/sec or kg/sec)
51-60	CTRB(J,6)	Influx load of contaminant associated with clay (pC/sec or kg/sec)
61-70	CTRB(J,7)	Influx load of dissolved contaminant (pC/sec or kg/sec)

Example:



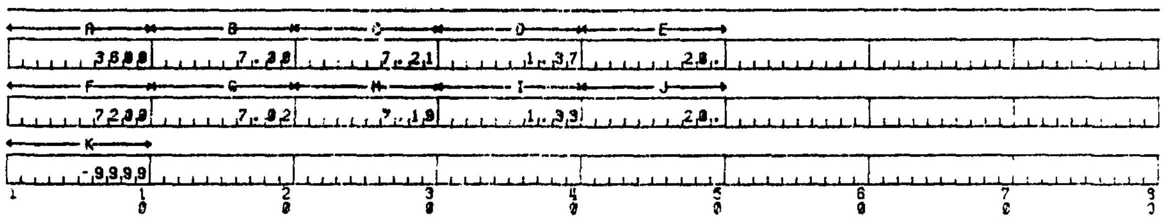
A) There is a tributary joining at this segment (NTRIBS = 1), and
B) the user will supply one value for each sediment and contaminant load and the model will distribute it vertically uniformly (TRBOPT = 0).
C) Use the following tributary data until 3600 seconds.
D) Influx load of sand from the tributary is 2.0 kg/sec.
E) Influx load of silt from the tributary is 5.0 kg/sec.
F) Influx load of clay from the tributary is 4.0 kg/sec.
G) Influx load of contaminant associated with sand is 1.0 x 10^{-7} kg/sec.
H) Influx load of contaminant associated with silt is 2.0 x 10^{-6} kg/sec.
I) Influx load of contaminant associated with clay is 1.0 x 10^{-5} kg/sec.
J) Influx load of dissolved contaminant is 2.0 x 10^{-7} kg/sec.
K) "-9999" signals the end of the tributary data.

Data Set 19 - Hydrologic Data

This data set is used to input discharge, depth and water temperature. This set is either time varying or not time varying. If it is time varying then as with the other time varying data types, the model treats these data as a step function.

Format: I10, 4F10.0

Column	Variable	Description
1-10	ENDTIM	Ending time of the data in seconds. This record will be used until the simulation passes this time plane. Then the next data set will be used. This variable is also used to signal the end of the hydrologic data by inputting a value of "-9999".
11-20	QI	Total discharge into this segment in m³/sec.
21-30	QO	Total discharge out of this segment in m³/sec
31-40	DEPTH	Flow depth in meters
41-50	TEMPR	Water temperature in centigrade.

Example:

There are two sets of hydrologic data in the above samples and the data on the first record will be used until one time step beyond.

A) 3600 seconds. During the time plane of 3600 seconds the discharge into the segment is
B) 7.00 m^3/sec; the discharge out of the segment is
C) 7.21 m^3/sec; with an average flow depth of
D) 1.37 meters; and the water temperature is
E) 20 C. The second set of data will be used from one time step beyond 3600 seconds until the end of the simulation which cannot exceed
F) 7200 seconds. The values have changed slightly from the first set. The discharge into the segment increases to
G) 7.02 m^3/sec; the discharge from the segments drops to
H) 7.19 m^3/sec; the flow depth decreases to
I) 1.33 meters, but the temperature remains at
J) 20 C. The end of the hydrologic data for this segment is signaled by
K) -9999.

SERATRA ERROR MESSAGES

Several checks for data correctness and consistency and run time error are built into SERATRA. The message

    *****SERATRA -- FATAL ERROR*****
    PRINT "SED.LST" FOR DETAILS

will be printed at the terminal device whenever a detected error prohibits further input processing or produces meaningless results. The reason the simulation is terminated will be reported to the file "SED.LST". The two basic groups of errors are input data processing errors and simulation run time errors. Some error messages are listed below.

DEPOSITION EXCEEDS PERMISSIBLE BED DEPTH IN BEDHST NBED = XXXXX

The total number of bed layers is limited to 10 in this model and exceeding the limit will cause this error to be printed by Subroutine BEDHST. If the amount of deposition is not unreasonable, the thickness of the bed layers will have to be increased.

*****FATAL ERROR -- SUBROUTINE COLBY*****

This message is immediately followed by another message that indicates which of the three limitations has been violated. When this happens, SERATRA computes the sand carrying capacity using the Toffaletti method and continues the simulation. The error is fatal only as far as Subroutine COLBY is concerned. The indications and ranges are:

DOUT	0.10 mm $\leq$ median bed material size $\leq$ 0.80 mm
ROUT	0.10 ft $\leq$ hydraulic radius $\leq$ 100.0 ft
VOUT	1.0 fps $\leq$ average velocity $\leq$ 10.0 fps

Another error message from Subroutine COLBY is

*****SUBROUTINE COLBY -- FLS WENT > 2.E + 5

Since the Colby method has a correction factor accounting for the fine sediment concentration up to 200,000 ppm, river flows carrying a fine sediment with more than 200,000 ppm will be out of the application range of the Colby method. However, this is not a fatal error and the simulation will continually use the Colby method.

The next three run time errors concern the time varying data sets of hydrologic data, initial conditions to Segment 1, and tributary data. They are written to SED.LST when the elapsed simulation time goes beyond the time range of any of these data sets. However, this should never happen because prior to simulating the segment, each available data set is examined and insufficient data should be discovered at that time. Hence, any appearance of one of the following errors means that something is wrong with the temporary files:

FATAL ERROR - INITIAL CONDITION DATA TO SEGMENT ONE HAS BEEN EXHAUSTED.
FATAL ERROR - HYDROLOGY DATA HAS BEEN EXHAUSTED.
FATAL ERROR - TRIBUTARY DATA HAS BEEN EXHAUSTED.

<u>Utility Program MSGENT</u>

SERATRA has a unique method of handling error messages in that each error is assigned a number that is a pointer into an error message data file. The information for each error consists of the type of error (warning or fatal), error number and the error message text.

The error message file ("SERATERR.MSG") is a direct access file and this program is needed to write the error information into the file from the source file ("MSGENT.DAT"). Of course it needs to be run only when the source file has been changed or new errors have been added. The file has a capacity of 100 separate errors and if this is exceeded, coding changes to MSGENT.FLX and SERATRA subroutine will be necessary.

The format of each error message (it is the same for both the source and direct access file) is as follows:

Record 1 - Format: A1, I3, 76A1

Column	Variable	Description
1	ERRTYP	Type of error message = W; warning error = F; fatal error. Processing discontinues.
2-4	ERRNUM	Error number
5-80	ERRMSG	First 76 characters of the error message text.

Record 2 - Format: 80A1

1-80	ERRMSG	Last 80 columns of the error message text.

SERATRA POST-PROCESSING ROUTINE (SPPR)

There are two reasons for writing the time plane results to a file rather than just listing them on a printer device: 1) result files are more flexible, and 2) long-term simulations can produce an excessively large printout. Because of these two main reasons, a file naming convention has been adopted that allows for a straightforward post processing.

The user supplies the basic file descriptor, which consists of a 5-digit name and a 3-digit extension (plus any nondefault devices or UIC's). SERATRA attaches a 4-digit river segment number to the 5-digit base name. For example, DB: 106,5 CALIB0010.RLT would refer to the 10th river segment of the base result file DB: 106,5 CALIB.RLT. This facilitates a chaining operation for postprocessing. To process the same time steps for all the river segments all that has to be specified is the base file name, and beginning and ending segment numbers.

SPPR will currently only list the results of the time plane, but it will eventually be capable of also plotting the results. The results that are reported are as follows:

1) River segment number
2) Time step number
3) Elapsed time
4) Datum
5) Number of elements

6) Standard element thickness
7) Top element thickness
8) Number of bed layers
9) Standard bed layer thickness
10) Top bed layer thickness
11) Shear stress value
12) Concentrations of sediment and contaminants in water columns
    a) Element elevations
    b) Sand concentration
    c) Silt concentration
    d) Clay concentration
    e) Dissolved contaminant concentration
    f) Concentration of contaminant with sand
    g) Concentration of contaminant with silt
    h) Concentration of contaminant with clay
    i) Average particulate contaminant concentration per unit weight of sediment
    j) Total particulate contaminant concentrations per unit volume of water
    k) Total contaminant concentration (sum of particulate and dissolved contaminant concentrations) per unit volume of water
13) Concentrations of sediment and contaminants in the river bed
    a) Bed layer elevation
    b) Sand weight fraction
    c) Silt weight fraction
    d) Clay weight fraction
    e) Concentration of particulate contaminant with sand per unit weight of sediment
    f) Concentration of particulate contaminant with silt per unit weight of sediment
    g) Concentration of particulate contaminant with clay per unit weight of sediment
    h) Average particulate contaminant concentration per unit weight of sediment

When SPPR is activated, it announces itself and then interrogates the user to learn the parameters of the session such as file name, segment number, and time plane numbers. The user's responses are underlined.

```
*****SERATRA POST PROCESSING*****
IS THIS TO BE A CHAINED OPERATION (Y OR N) > Y
```

The user must respond with an "N" if the processing is to be restricted to only one river segment. The program will ask for the complete name including the river segment number of the result file to be processed. Otherwise enter "Y".

```
ENTER THE NAME OF THE FILE TO BE PROCESSED > MT1:3YEAR0003.RLT
```

Responding to the previous question with an "Y" indicates that the processing will be performed on two or more river segments and SPPR will need to know the base file name, beginning segment, ending segment and the interval between segment numbers.

```
ENTER BASE FILE NAME > MT1:3YEAR.RLT
ENTER BEGINNING SEGMENT NUMBER (I4) > 2
ENTER ENDING SEGMENT NUMBER (I4) > 10
ENTER INTERVAL BETWEEN SEGMENTS (I4) > 4
```

The information above tells SPPR to process the following files:

1) MT1:3YEAR0002.RLT
2) MT1:3YEAR0006.RLT
3) MT1:3YEAR0010.RLT

The next set of information specifies which time planes will be processed.

```
ENTER BEGINNING TIME PLANE NUMBER (I10) > 2
ENTER ENDING TIME PLANE NUMBER (I10) > 6
ENTER INTERVAL BETWEEN TIME PLANES (I10) > 2
ENTER TIME STEP SIZE (F10.0) > 0.5
```

SPPR will interpolate the information above and will process the second, fourth and sixth time step written to the file. Note that the time plane numbers being asked for are the sequence numbers of the data in the file and not the original time plane number used for the simulation. If during the simulation, every fourth time plane was written to the result file, the specifications above would cause time plane numbers 8, 16, and 24 to be processed.

The assigned time step size in hours must be the same as the simulation time step or the multiplication of the simulation time step.

The final input concerns the units of the contaminant that was modeled.

```
ENTER THE CONCENTRATION UNITS (PC or KG) > KG
```

If radionuclides are modeled, the proper response would be "PC" and for pesticides and other chemicals the user should answer "KG".

SECTION 4

REFERENCES

Onishi, Y., P. A. Johanson, R. G Baca and E. L. Hilty. 1976. Studies of Columbia River Water Quality - Development of Mathematical Models for Sediment and Radionuclide Transport Analysis. BNWL-B-452, Pacific Northwest Laboratory, Richland, WA.

Onishi, Y. 1977. Mathematical Simulation of Sediment and Radionuclide Transport in the Columbia River. BNWL-2228, Pacific Northwest Laboratory, Richland, WA.

Onishi, Y., D. L. Schreiber and R. B. Codell. 1979a. "Mathematical Simulation of Sediment and Radionuclide Transport in the Clinch River, Tennessee." Proceedings of ACS/CSJ Chemical Congress, Honolulu, Hawaii, April 1-6, 1979. Contaminants and Sediments. R. A. Baker (ed.), Ann Arbor Science Publishers, Inc., Ann Arbor, MI.

Onishi, Y., S. M. Brown, A. R. Olsen, M. A. Parkhurst, S. E. Wise and W. H. Walters. 1979b. Methodology for Overland and Instream Migration and Risk Assessment of Pesticides. Battelle, Pacific Northwest Laboratories, Richland, WA.

Onishi, Y. and S. E. Wise. 1979. Mathematical Model, SERATRA, for Sediment and Contaminant Transport in Rivers and its Application to Pesticide Transport in Four Mile and Wolf Creeks in Iowa. Battelle, Pacific Northwest Laboratories, Richland, WA.

Smith, J. H., W. R. Mabey, N. Bohonos, B. R. Holt, S. S. Lee, T. W. Chou, D. C. Bomberger, and T. Mill. 1977. Environmental Pathways of Selected Chemicals in Freshwater Systems. EPA-600/7-77-113.

Stanford Research Institute. 1979. Fate of Selected Pollutants -- Protocall II. Photolysis in Water. Environmental Protection Agency Contract Number 68-03-2227.

APPENDIX - SERATRA LISTING

(FLECS VERSION 22.46)   13-MAR-81   13:17:59 PAGE 00001

```
        ----------------------------------------
              SUBROUTINE BEDDAT(B,ECHO,NBED,AREA,BDIV,DENS,POR,XYSO,TBED)
        C
        C    THIS ROUTINE IS RESPONSIBLE FOR READING THE INITIAL BED CONDITIONS.
        C
        C      FORMAL PARAMETERS:
        C        B     - INITIAL BED CONDITIONS
        C        ECHO  - LINE PRINTER ECHO OPTION CONTROL VARIABLE (L*1)
        C        NBED  - NUMBER OF BED LAYERS
        C      AREA - VERTICAL PROJECTION AREA
        C      BDIV - THICKNESS OF BED ELEMENTS
        C      DENS - DENSITY
        C      POR  - POROSITY
        C      XYSO - THICKNESS OF TOP BED ELEMENT
        C      TBED - WEIGHT OF MATERIAL, TOTAL OF CONTAMINANT IN BED
        C
        C    CALLED BY: SERATRA
        C
              INCLUDE 'ELMSIZ.PRM'
        C     INCLUDE 'SY:ELMSIZ.PRM'
        C
              LOGICAL*1 ECHO
                INTEGER SWITCH
        C
              DIMENSION B(MAXLEV,MAXCON-1), AREA(MXELEM), DENS(3), TBED(MAXCON),
             1          CELL(MAXLEV,MAXCON)
        C
        C    CARD 8....INITIAL BED CONDITIONS
        C
        C      A SET OF INITIAL BED CONDITIONS ARE READ FOR EACH THE SIX
        C      PARAMETERS (LAYERS ARE NUMBERED BEGINNING AT THE BOTTOM,
        C      (LAYER 1), AND ENDING WITH THE SURFACE LAYER, (LAYER #NBED)),
        C
        C      IF THERE IS NO VERTICAL VARIATION, COLUMNS 1-5 CONTAIN A NEGATIVE
        C      VALUE AND COLUMNS 6-15 CONTAIN THE CONSTANT VALUE. WHEN THE DATA
        C      DOES VARY WITH DEPTH, A VALUE IS READ FOR EACH ELEMENT. THE UNITS
        C      OF THE CONTAMINANT CONCENTRATIONS DEPEND UPON THE TYPE OF CONTAMINANT
        C      (RADIONUCLIDE..PC/KG OR PESTICIDE..KG/KG).
        C
        C                  PARAMETERS ARE READ IN THE FOLLOWING ORDER:
        C
        C         PARAMETER 1...WEIGHT FRACTION OF SAND IN THE BED
        C                   2...WEIGHT FRACTION OF SILT IN THE BED
        C                   3...WEIGHT FRACTION OF CLAY IN THE BED
        C                   4...CONTAMINANT CONCENTRATION IN SAND
        C                   5...CONTAMINANT CONCENTRATION IN SILT
        C                   6...CONTAMINANT CONCENTRATION IN CLAY
        C
              DO (K=1,MAXCON-1)
              .  READ(1,2) SWITCH,VALUE
              .  WHEN (SWITCH .LT. 0)
        C     .  .  *** PARAMETER DOES NOT VARY VERTICALLY ***
              .  .  DO (J=1,NBED) B(J,K) = VALUE
              .  ...FIN
```

```
         .  ELSE
C        .  .  *** PARAMETER VARIES VERTICALLY ***
         .  .  READ(1,1) (B(J,K),J=1,NBED)
         .  ...FIN
         ...FIN
         IF (ECHO)
         .  WRITE(6,3)
         .  DO (J=1,NBED)
         .  .  WRITE(6,4) J,(B(J,K),K=1,MAXCON-1)
         .  ...FIN
         ...FIN
C
         DO (J=1,MAXCON) TBED(J)=0.
         DO (J=1,MAXCON-1)
         .  DO (I=1,NBED)
         .  .  WHEN (J.LT.4)
         .  .  .  DNSITY = (1.-POR)/(B(I,1)/DENS(1)+B(I,2)/DENS(2)+
        1.  .  .           B(I,3)/DENS(3))
         .  .  .  DEL = BDIV
         .  .  .  IF (I.EQ.NBED) DEL = XYSO
         .  .  .  VOL=DEL*AREA(1)
         .  .  .  CELL(I,J) = B(I,J)*VOL*DNSITY
         .  .  ...FIN
         .  .  ELSE CELL(I,J) = CELL(I,J-3)*B(I,J)
         .  .  TBED(J) = TBED(J) + CELL(I,J)
         .  ...FIN
         ...FIN
         DO (J = 4,MAXCON-1)
         .  TBED(MAXCON) = TBED (MAXCON) + TBED(J)
         ...FIN
         RETURN
C
 1       FORMAT(8F10.0)
 2       FORMAT(I5,F10.0)
 3       FORMAT(1H0,53X,'INITIAL BED CONDITIONS'/1H0,'LAYER',
        1  3(3X,'WEIGHT FRACTION',2X),3(1X,'CONTAMINANT CONC.',2X)/
        2  14X,'IN SAND',13X,'IN SILT',13X,'IN CLAY',
        3  12X,'IN SAND',13X,'IN SILT',13X,'IN CLAY')
 4       FORMAT(2X,I2,2(7X,1PE12.5,2(8X,1PE12.5)))
C
         END
```

(FLECS VERSION 22.46)

```
(FLECS VERSION 22.46)     13-MAR-81    13:18:18  PAGE  00001

        ----------------------------------------

        SUBROUTINE BEDDK(B, DECAY, DELTD, NBED)
C
C   THIS ROUTINE CALCULATES THE DECAY OF THE CONTAMINANTS IN THE
C   RIVER BED.
C
C     FORMAL PARAMETERS:
C       B       - BED CONCENTRATIONS
C       DECAY   - DECAY VALUES
C       DELTD   - TIME STEP IN DAYS
C       NBED    - NUMBER OF BED LAYERS
C
C   CALLED BY: TRANSP, SERATRA
C
        INCLUDE 'SY:ELMSIZ.PRM'
C
        DIMENSION B(MAXLEV,MAXCON-1), DECAY(6)
C
        IF(DECAY(1) .NE. 0.0)
        .  DO (JJ=4,6)
C       .  .  *** RADIONUCLIDE DECAY ***
        .  .  DO (JK=1,NBED)
        .  .  .  B(JK,JJ) = B(JK,JJ)*EXP(-DECAY(1)*DELTD)
        .  .  ...FIN
        .  ...FIN
        ...FIN
        RETURN
        END

        (FLECS VERSION 22.46)

        ----------------------------------------
```

```
          ----------------------------------------

          SUBROUTINE BEDHIS(B, BDIV, BED, COLD, DELTD, DELZ,  DENS,
         1             FERROR, ILAYR, NBED, NELEM, POR, XNT, XYSO,
         2             DEPO, SCOUR, BEDSD)
C
C         THIS SUBROUTINE KEEPS A RECORD OF BED HISTORY, INCLUDING BED
C          SURFACE ELEVATION, RATIO OF BED SEDIMENT WEIGHT FRACTIONS, AND
C          ASSOCIATED CONCENTRATIONS IN THE BED
C
C         FORMAL PARAMETERS:
C            B       - BED CONDITIONS (WEIGHT FRACTION, PC/KG)
C            BDIV    - STANDARD BED LAYER THICKNESS (M)
C            BED     - BED THICKNESS (M)
C         BEDSD  - TRANSFER OF DISSOLVED TO ABSORPED (PC/M2/DAY)
C            COLD    - CELL-CENTERED CONCENTRATION (KG/M**3,PC/M**3)
C            DELTD   - TIME STEP (DAYS)
C            DELZ    - STANDARD ELEMENT THICKNESS (M)
C            DENS    - DENSITY (KG/M**3)
C         DEPO    - DEPOSITION RATE (KG[PC]/M2/DAY)
C            FERROR - FATAL ERROR FLAG (L*1)
C            ILAYR   - NO. OF LAYERS COMPLETELY SCOURED BY EACH RESPECTIVE
C                        SEDIMENT, ILAYR(J)=-1 FOR DEPOSITION
C            NBED    - NUMBER OF BED LAYERS
C            NELEM   - NUMBER OF ELEMENTS
C            POR     - POROSITY
C         SCOUR   - SCOUR RATE (KG[PC]/M2/DAY)
C            XNT     - WEIGHT OF THE BED SEDIMENT LAYER (KG/M2)
C            XYSO    - THICKNESS OF TOP BED LAYER (M)
C         ZERO    - NORMALIZED TRUNCATION ERROR - SIGNIFICANT DIGITS
C
C         CALLED BY: TRANSP
C
          INCLUDE 'SY:ELMSIZ.PRM'
C
          LOGICAL*1 FERROR
C
          DIMENSION ALEFT(3), ARAD(3), B(MAXLEV,MAXCON-1), B2(6),
         1            BEDSD(3), COLD(MXELEM,MAXCON), DENS(3), DEPO(6),
         2               ILAYR(3), SCOUR(6), SUMSD(3), SUMSDC(3), XNT(3)
          DATA ZERO/1.0E-8/
C
          FERROR = .FALSE.
C
          IN=ILAYR(1)
          IP=ILAYR(2)
          IQ=ILAYR(3)
C
          DO (IJ=1,3)
          .  SUMSD(IJ)=DEPO(IJ)
          .  SUMSDC(IJ)=DEPO(IJ+3)
          .  IF(BEDSD(IJ).LT.0.0) SUMSDC(IJ)=-BEDSD(IJ)+SUMSDC(IJ)
          ...FIN
C           ARAD  -- AMOUNT OF CHEMICALS LEFT IN TOP BED LAYER.
C         ALEFT  -- AMOUNT OF SEDIMENT LEFT IN TOP BED LAYER.
```

```
C
        DO (II=1,3)
        .  ARAD(II)=0.0
        .  ALEFT(II)=0.0
        ...FIN
C
        IF(SCOUR(1)+SCOUR(2)+SCOUR(3).GT.0.0) GO TO 110
        TMP=(XNT(1)/DENS(1)+XNT(2)/DENS(2)+XNT(3)/DENS(3))/
       1    (1.0-POR)
        TDEPO=DEPO(1)+DEPO(2)+DEPO(3)
        DEL=(TMP - BDIV)/BDIV
        IF(TDEPO.GT.0.0.AND.ABS(DEL).LE.ZERO)
C*************************************************************
C       .  IF ONLY DEPOSITION OCCURS AND THE TOP LAYER HAS A    *
C       .  THICKNESS OF BDIV...TO AVOID HOMOGENIZING THE OLD    *
C       .  AND NEW MATERIAL WE CREATE A NEW ELEMENT WITH        *
C       .  XNT(I) = 0, AND ALEFT(I) AND ARAD(I) EQUAL TO        *
C       .  DEPOSITED MATERIAL AND CONTAMINANT RESPECTIVELY.     *
C*************************************************************
        .  XNT(1)=0.
        .  XNT(2)=0.
        .  XNT(3)=0.
        .  NBED=NBED+1
        .  DO (I=1,3)
        .  .  ALEFT(I)=SUMSD(I)*DELTD
        .  .  ARAD(I)=SUMSDC(I)*DELTD
        .  .  IF(BEDSD(I).GT.0.0) ARAD(I)=ARAD(I)-BEDSD(I)*DELTD
        .  ...FIN
        .  GO TO 270
        ...FIN
C************************************************************
C       COMPUTES SEDIMENT (KG/M2) AND CONTAMINANT (PC/M2)  *
C       RESIDING IN THE TOP LAYER                          *
C************************************************************
 110    IF (IN.LT.0) IN=0
        ALEFT(1)=XNT(1)+SUMSD(1)*DELTD
        ARAD(1)=XNT(1)*B(NBED-IN,4)+SUMSDC(1)*DELTD
        IF (BEDSD(1).GT.0.0) ARAD(1)=ARAD(1)-BEDSD(1)*DELTD
        IN=ILAYR(1)
        IF (IP.LT.0) IP=0
        ALEFT(2)=XNT(2)+SUMSD(2)*DELTD
        ARAD(2)=XNT(2)*B(NBED-IP,5)+SUMSDC(2)*DELTD
        IF(BEDSD(2).GT.0.0) ARAD(2)=ARAD(2)-BEDSD(2)*DELTD
        IP = ILAYR(2)
        IF (IQ .LT. 0) IQ=0
        ALEFT(3)=XNT(3)+SUMSD(3)*DELTD
        ARAD(3)=XNT(3)*B(NBED-IQ,6)+SUMSDC(3)*DELTD
        IF(BEDSD(3).GT.0.0) ARAD(3)=ARAD(3)-BEDSD(3)*DELTD
        IQ = ILAYR(3)
C***********************************************************
C       IF SAND HAS NOT SCOURED A COMPLETE LAYER ALEFT(I) *
C       AND ARAD(I) ARE COMPLETELY DETERMINED             *
C***********************************************************
        IF (IN.LT.1) GO TO 270
C***********************************************************
C       IF SILT AND SAND EROSION (DEPOSITION) ARE WITHIN  *
```

```
C      THE SAME LAYER ... ALEFT(2), AND ARAD(2) ARE              *
C      COMPLETELY DETERMINED ...                                 *
C      OTHERWISE INCLUDE ADDITIONAL LAYERS                       *
C*****************************************************************
       IF(IN.GT.IP)
       .  IPP = IP +1
       .  IF (IPP.EQ.0) IPP = 1
       .  DO (IT=IPP,IN)
       .  .  NB = NBED - IT
       .  .  XND = (1.0-POR)/(B(NB,1)/DENS(1)+B(NB,2)/DENS(2)+
      1.  .        B(NB,3)/DENS(3))
       .  .  DELXNT=XND*BDIV*B(NB,2)
       .  .  ALEFT(2)=ALEFT(2) + DELXNT
       .  .  ARAD(2)=ARAD(2) + DELXNT*B(NB,5)
       .  ...FIN
       ...FIN
C******************************************************************
C      IF CLAY AND SAND EROSION (DEPOSITION) ARE WITHIN    *
C      THE SAME LAYER ... ALEFT(3) AND ARAD(3) ARE         *
C      COMPLETELY DETERMINED....                           *
C      OTHERWISE INCLUDE ADDITIONAL LAYERS                 *
C******************************************************************
       IF(IN.GT.IQ)
       .  IQQ = IQ + 1
       .  IF (IQQ.EQ.0) IQQ = 1
       .  DO (IT=IQQ,IN)
       .  .  NB = NBED - IT
       .  .  XND = (1.0-POR)/(B(NB,1)/DENS(1)+B(NB,2)/DENS(2)+
      1.  .        B(NB,3)/DENS(3))
       .  .  DELXNT=XND*BDIV*B(NB,3)
       .  .  ALEFT(3) = ALEFT(3) + DELXNT
       .  .  ARAD(3) = ARAD(3) + DELXNT*B(NB,5)
       .  ...FIN
       ...FIN
C******************************************************************
C      ESTABLISH THE B MATRIX VALUES FOR THE NEWLY CREATED *
C      BED ELEMENTS                                        *
C******************************************************************
270    CONTINUE
       B1 = ALEFT(1) + ALEFT(2) + ALEFT(3)
       XM = (ALEFT(1)/DENS(1) + ALEFT(2)/DENS(2) + ALEFT(3)/
      1       DENS(3))/(1.0 - POR)
       IW = XM/BDIV
       REMAIN =XM - IW*BDIV
       IF(REMAIN.GT.ZERO) IW =IW +1
       NBED =NBED - IN - 1
       IF (IN.LT.0) NBED =NBED-1
       NBED1 = NBED + 1
       NBED2 = NBED + IW
       DO (IX=1,3)
       .  B2(IX) = ALEFT(IX)/B1
       .  B2(IX+3) = 0.0
       .  IF(B2(IX).GT.ZERO) B2(IX+3) = ARAD(IX)/ALEFT(IX)
       ...FIN
       DO (IY=NBED1,NBED2)
       .  DO (IX=1,6)
```

```
         .  .  B(IY,IX) = B2(IX)
         .  ...FIN
         ...FIN
         NBED = NBED + IW
         XYSO = REMAIN
         IF (REMAIN.LE.ZERO) XYSO = BDIV
         BED = (NBED-1) * BDIV + XYSO
         IF (NBED .GT. MAXLEV)
         .  WRITE(6,200) NBED
   200   .  FORMAT(2X,'DEPOSITION EXCEEDS PERMISSIBLE BED DEPTH IN BEDHST',/,
        1.  5X,'NBED=',I5)
         .  FERROR = .TRUE.
         ...FIN
   C
         RETURN
         END

   (FLECS VERSION 22.46)

   ------------------------------------
```

```
----------------------------------------
      SUBROUTINE COLBY(ALEN, C, DELZ,  D50, HRAD, NELEM, QTOT,
     1                 TEMPR, VOL, GSI, FERROR)
C
C     THIS SUBROUTINE USES COLBY'S METHOD TO CALCULATE THE CAPACITY OF
C     THE FLOW TO TRANSPORT SAND.
C
C        INPUT PARAMETERS:
C          ALEN    - SEGMENT LENGTH
C          C       - NODAL VALUES OF CONCENTRATION
C          DELZ    - STANDARD ELEMENT THICKNESS
C          D50     - MEDIAN BED SEDIMENT DIAMETER (M)
C          HRAD    - HYDRAULIC RADIUS
C          NELEM   - NUMBER OF VERTICAL ELEMENTS
C          QTOT    - TOTAL FLOW WITHIN THE SEGMENT
C          TEMPR   - WATER TEMPERATURE
C          V       - AVERAGE VELOCITY
C          VOL     - VOLUME
C
C        OUTPUT PARAMETERS:
C          GSI     - TOTAL SAND TRANSPORT
C          FERROR - FATAL ERROR FLAG (L*1)
C
C     CALLED BY: SAND
C
C     THE COLBY METHOD HAS THE FOLLOWING UNITS AND APPLICABLE RANGES OF
C     VARIABLES.
C        AVERAGE VELOCITY..............V.......FPS.........1-10 FPS
C        HYDRAULIC RADIUS..............FHRAD...FT..........1-100 FT
C        WATER SURFACE WIDTH...........W.......FT..........
C        MEDIAN BED MATERIAL SIZE......DB50....MM..........0.1-0.8 MM
C        TEMPERATURE...................TMPR....DEG F.......32-100 DEG.
C        FINE SEDIMENT CONCENTRATION...FSL.....MG/LITER....0-200000 PPM
C        TOTAL SEDIMENT LOAD...........GSI.....TON.........
C
      INCLUDE 'SY:ELMSIZ.PRM'
C
      LOGICAL*1 FERROR
C
      DIMENSION C(MXELEM,MAXCON),CF(5),DF(10),DG(4),DP(11),D50G(6),
     1          F(5,10), G(4,8,6), II(2), JJ(2), KK(2), P(11), T(7,4),
     2          TEMP(7), VG(8), X(2,2), XA(2), XCT(2), XF(2,2),
     3          XG(2), XT(2,2), XX(2), YY(2), ZZ(2)
C
      DATA G(1,1,1),G(2,1,1),G(3,1,1),G(4,1,1)/1.0, 0.30, 0.06, 0.00/
      DATA G(1,2,1),G(2,2,1),G(3,2,1),G(4,2,1)/3.00, 3.30, 2.50, 2.00/
      DATA G(1,3,1),G(2,3,1),G(3,3,1),G(4,3,1)/5.40, 9.0, 10.0, 20.0/
      DATA G(1,4,1),G(2,4,1),G(3,4,1),G(4,4,1)/11.0, 26.0, 50.0,150.0/
      DATA G(1,5,1),G(2,5,1),G(3,5,1),G(4,5,1)/17., 49., 130., 500./
      DATA G(1,6,1),G(2,6,1),G(3,6,1),G(4,6,1)/29., 101., 400., 1350./
      DATA G(1,7,1),G(2,7,1),G(3,7,1),G(4,7,1)/44.,160.,700.,2500./
      DATA G(1,8,1),G(2,8,1),G(3,8,1),G(4,8,1)/60.,220.,1000.,4400./
      DATA G(1,1,2),G(2,1,2),G(3,1,2),G(4,1,2)/0.38, 0.06, 0.0, 0.0/
      DATA G(1,2,2),G(2,2,2),G(3,2,2),G(4,2,2)/1.60, 1.20, 0.65, 0.10/
```

```
          DATA G(1,3,2),G(2,3,2),G(3,3,2),G(4,3,2)/3.70, 5., 4., 3./
          DATA G(1,4,2),G(2,4,2),G(3,4,2),G(4,4,2)/10., 18., 30., 52./
          DATA G(1,5,2),G(2,5,2),G(3,5,2),G(4,5,2)/17., 40., 80., 160./
          DATA G(1,6,2),G(2,6,2),G(3,6,2),G(4,6,2)/36., 95., 230., 650./
          DATA G(1,7,2),G(2,7,2),G(3,7,2),G(4,7,2)/60., 150., 415., 1200./
          DATA G(1,8,2),G(2,8,2),G(3,8,2),G(4,8,2)/81., 215., 620., 1500./
          DATA G(1,1,3),G(2,1,3),G(3,1,3),G(4,1,3)/0.14, 0.0, 0.0, 0.0/
          DATA G(1,2,3),G(2,2,3),G(3,2,3),G(4,2,3)/1., 0.60, 0.15, 0.0/
          DATA G(1,3,3),G(2,3,3),G(3,3,3),G(4,3,3)/3.30, 3.00, 1.70, 0.50/
          DATA G(1,4,3),G(2,4,3),G(3,4,3),G(4,4,3)/11., 15., 17., 14./
          DATA G(1,5,3),G(2,5,3),G(3,5,3),G(4,5,3)/20., 35., 49., 70./
          DATA G(1,6,3),G(2,6,3),G(3,6,3),G(4,6,3)/44., 85., 150., 250./
          DATA G(1,7,3),G(2,7,3),G(3,7,3),G(4,7,3)/71., 145., 290., 500./
          DATA G(1,8,3),G(2,8,3),G(3,8,3),G(4,8,3)/100., 202., 400., 700./
          DATA G(1,1,4),G(2,1,4),G(3,1,4),G(4,1,4)/0.0, 0.0, 0.0, 0.0/
          DATA G(1,2,4),G(2,2,4),G(3,2,4),G(4,2,4)/0.70, 0.30, 0.06, 0.0/
          DATA G(1,3,4),G(2,3,4),G(3,3,4),G(4,3,4)/2.9, 2.3, 1.0, 0.06/
          DATA G(1,4,4),G(2,4,4),G(3,4,4),G(4,4,4)/11.5, 13., 12., 7./
          DATA G(1,5,4),G(2,5,4),G(3,5,4),G(4,5,4)/22., 31., 40., 50./
          DATA G(1,6,4),G(2,6,4),G(3,6,4),G(4,6,4)/47., 84., 135., 210./
          DATA G(1,7,4),G(2,7,4),G(3,7,4),G(4,7,4)/75., 140., 240., 410./
          DATA G(1,8,4),G(2,8,4),G(3,8,4),G(4,8,4)/106, 190., 350., 630./
          DATA G(1,1,5),G(2,1,5),G(3,1,5),G(4,1,5)/0.0, 0.0, 0.0, 0.0/
          DATA G(1,2,5),G(2,2,5),G(3,2,5),G(4,2,5)/0.44, 0.06, 0.0, 0.0/
          DATA G(1,3,5),G(2,3,5),G(3,3,5),G(4,3,5)/2.8, 1.8, 0.6, 0.0/
          DATA G(1,4,5),G(2,4,5),G(3,4,5),G(4,4,5)/12., 12.5, 10., 4.5/
          DATA G(1,5,5),G(2,5,5),G(3,5,5),G(4,5,5)/24., 30., 35., 37./
          DATA G(1,6,5),G(2,6,5),G(3,6,5),G(4,6,5)/52., 78., 120., 190./
          DATA G(1,7,5),G(2,7,5),G(3,7,5),G(4,7,5)/83., 180., 215., 380./
          DATA G(1,8,5),G(2,8,5),G(3,8,5),G(4,8,5)/120., 190., 305., 550./
          DATA G(1,1,6),G(2,1,6),G(3,1,6),G(4,1,6)/0.0, 0.0, 0.0, 0.0/
          DATA G(1,2,6),G(2,2,6),G(3,2,6),G(4,2,6)/0.3, 0.0, 0.0, 0.0/
          DATA G(1,3,6),G(2,3,6),G(3,3,6),G(4,3,6)/2.9, 1.4, 0.3, 0.0/
          DATA G(1,4,6),G(2,4,6),G(3,4,6),G(4,4,6)/14., 11., 7.7, 3.0/
          DATA G(1,5,6),G(2,5,6),G(3,5,6),G(4,5,6)/27., 29., 30., 30./
          DATA G(1,6,6),G(2,6,6),G(3,6,6),G(4,6,6)/57., 75., 110., 170./
          DATA G(1,7,6),G(2,7,6),G(3,7,6),G(4,7,6)/90., 140., 200., 330./
          DATA G(1,8,6),G(2,8,6),G(3,8,6),G(4,8,6)/135., 190., 290., 520./
    C
          DATA F(1,1),F(1,2),F(1,3),F(1,4),F(1,5)/1., 1.1, 1.6, 2.6, 4.2/
          DATA F(1,2),F(2,2),F(3,2),F(4,2),F(5,2)/1., 1.1, 1.65, 2.75, 4.9/
          DATA F(1,3),F(2,3),F(3,3),F(4,3),F(5,3)/1., 1.1, 1.7, 3., 5.5/
          DATA F(1,4),F(2,4),F(3,4),F(4,4),F(5,4)/1., 1.12, 1.9, 3.6, 7./
          DATA F(1,5),F(2,5),F(3,5),F(4,5),F(5,5)/1., 1.17, 2.05, 4.3, 8.7/
          DATA F(1,6),F(2,6),F(3,6),F(4,6),F(5,6)/1., 1.2, 2.3, 5.5, 11.2/
          DATA F(1,7),F(2,7),F(3,7),F(4,7),F(5,7)/1., 1.22, 2.75, 8., 22./
          DATA F(1,8),F(2,8),F(3,8),F(4,8),F(5,8)/1., 1.25, 3., 9.6, 29./
          DATA F(1,9),F(2,9),F(3,9),F(4,9),F(5,9)/1., 1.3, 3.5, 12., 43./
          DATA F(1,10),F(2,10),F(3,10),F(4,10),F(5,10)/1.,1.4,4.9,22.,120./
    C
          DATA T /1.2, 1.15, 1.10, 0.96, 0.90, 0.85, 0.82, 1.35, 1.25,
         1        1.12, 0.92, 0.86, 0.80, 0.75, 1.60, 1.40, 1.20, 0.89,
         2        0.80, 0.72, 0.66, 2.00, 1.65, 1.30, 0.85, 0.72, 0.63,
         3        0.55/
    C
          DATA DF /0.10, 0.20, 0.30, 0.60, 1.00, 2.00, 6.00, 10.00,
```

```
      1          20.00, 1.E2/
C
       DATA CF /9.00, 1.E4, 5.E4, 1.E5, 1.5E5/
C
       DATA P /0.60, 0.90, 1.0, 1.0, 0.83, 0.60, 0.40, 0.25, 0.15,
      1        0.09, 0.05/
C
       DATA DP /0.10, 0.15, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70,
      1          0.80, 0.90, 1.00/
C
       DATA DG /0.10, 1.00, 10.0, 100./
C
       DATA VG/1.0, 1.5, 2.0, 3.0, 4.0, 6.0, 8.0, 10./
C
       DATA D50G/0.10, 0.20, 0.30, 0.40, 0.60, 0.80/
C
       DATA TEMP/32., 40., 50., 70., 80., 90., 100./
       DB50 = D50 * 1000.0
       FHRAD = HRAD * 3.280833
       TMPR = TEMPR * 1.8 + 32.0
       V = (QTOT * ALEN/VOL) * 3.7975E-5
       W = VOL/(HRAD*ALEN) * 3.280833
C      *** FSL....FINE SEDIMENT (I.E. COHESIVE SEDIMENT OR WASH) LOAD
C          IN MICRO GRAMS/LITER ***
       FSL = 0.0
       DO (IX=1,NELEM)
       .  FSL = FSL + 0.5*(C(IX,2) + C(IX+1,2) + C(IX,3) + C(IX+1,3))
       ...FIN
       FSL = FSL/NELEM * 1000.0
C
       IF((DB50 .LT. D50G(1)) .OR. (DB50 .GT. D50G(6)))
       .  FERROR = .TRUE.
       .  WRITE(6,1)
1      .  FORMAT(//10X,'***** FATAL ERROR -- SUBROUTINE COLBY *****')
       .  WRITE(6,2)
2      .  FORMAT(10X,'*****    DOUT    *****')
       ...FIN
       IF((FHRAD .LT. DG(1)) .OR. (FHRAD .GT. DG(4)))
       .  FERROR = .TRUE.
       .  WRITE(6,1)
       .  WRITE(6,3)
3      .  FORMAT(10X,'*****    ROUT    *****')
       ...FIN
       IF((V .LT. VG(1)) .OR. (V .GT. VG(8)))
       .  FERROR = .TRUE.
       .  WRITE(6,1)
       .  WRITE(6,4) V
4      .  FORMAT(10X,'*****    VOUT    *****',F10.5)
       ...FIN
       UNLESS (FERROR)
       .  IF(TMPR .LT. 32.0 .OR. TMPR .GT. 100.0)
       .  .  TMPR = 32.0
       .  ...FIN
       .  ID1 = 0
       .  ID2 = 0
       .  DO (I=1,3)
```

```
          .  .  IF ((FHRAD .GE. DG(I)) .AND. (FHRAD .LE. DG(I+1)))
          .  .  .  ID1 = I
          .  .  .  ID2 = I+1
          .  .  .  GO TO 114
          .  .  ...FIN
          .  ...FIN
    114   .  CONTINUE
          .  IV1 = 0
          .  IV2 = 0
          .  DO (I=1,7)
          .  .  IF ((V .GE. VG(I)) .AND. (V .LE. VG(I+1)))
          .  .  .  IV1 = I
          .  .  .  IV2 = I+1
          .  .  .  GO TO 118
          .  .  ...FIN
          .  ...FIN
    118   .  CONTINUE
          .  ID501 = 0
          .  ID502 = 0
          .  DO (I=1,5)
          .  .  IF ((DB50 .GE. D50G(I)) .AND. (DB50 .LE. D50G(I+1)))
          .  .  .  ID501 = I
          .  .  .  ID502 = I+1
          .  .  .  GO TO 122
          .  .  ...FIN
          .  ...FIN
    122   .  CONTINUE
          .  II(1) = ID1
          .  II(2) = ID2
          .  JJ(1) = IV1
          .  JJ(2) = IV2
          .  KK(1) = ID501
          .  KK(2) = ID502
          .  DO (I=1,2)
          .  .  I1 = II(I)
          .  .  XX(I) = ALOG10(DG(I1))
          .  .  DO (J=1,2)
          .  .  .  J1 = JJ(J)
          .  .  .  YY(J) = ALOG10(VG(J1))
          .  .  .  DO (K=1,2)
          .  .  .  .  K1 = KK(K)
          .  .  .  .  ZZ(K) = ALOG10(D50G(K1))
          .  .  .  .  IF (G(I1,J1,K1)-0.) 123,123,127
    123   .  .  .  .  CONTINUE
          .  .  .  .  DO (J3=J1,7)
          .  .  .  .  .  IF (G(I1,J3,K1)-0.) 124,124,126
    124   .  .  .  .  .  CONTINUE
          .  .  .  .  ...FIN
    126   .  .  .  .  CONTINUE
          .  .  .  .  X(J,K) = ALOG10(G(I1,J3,K1))+(ALOG10(VG(J1)/VG(J3)))*
        1.  .  .  .           (ALOG10(G(I1,J3+1,K1)/G(I1,J3,K1)))/(ALOG10(VG(J3+1)/
        2.  .  .  .           VG(J3)))
          .  .  .  .  GO TO 128
    127   .  .  .  .  CONTINUE
          .  .  .  .  X(J,K) = ALOG10(G(I1,J1,K1))
    128   .  .  .  .  CONTINUE
```

```
              .  .  .  ...FIN
              .  .  ...FIN
              .  .  XD = ALOG10(DB50) - ZZ(1)
              .  .  XN1 = X(1,2) - X(1,1)
              .  .  XN2 = X(2,2) - X(2,1)
              .  .  XDEN = ZZ(2) - ZZ(1)
              .  .  XA(1) = X(1,1) + XN1*XD/XDEN
              .  .  XA(2) = X(2,1) + XN2*XD/XDEN
              .  .  XNM = XA(2) - XA(1)
              .  .  XDY = YY(2) - YY(1)
              .  .  XG(1) = XA(1) + XNM*XV/XDY
              .  ...FIN
              .  XNM = XG(2) - XG(1)
              .  XD = ALOG10(FHRAD) - XX(1)
              .  XDEN = XX(2) - XX(1)
              .  GTUC = XG(1) + XNM*XD/XDEN
              .  GTUC = 10.**GTUC
      C       .
      C       .  *** GTUC IS UNCORRECTED GT IN LB/SEC/FT ***
      C       .
      C       .  *** NEXT APPLY FINE SEDIMENT LOAD AND TEMPERATURE CORRECTIONS ***
      C       .
              .  WHEN (TMPR .EQ. 60.) CFT = 1.0
              .  ELSE
              .  .  IT1 = 0
              .  .  IT2 = 0
              .  .  DO (I=1,6)
              .  .  .  IF ((TMPR .GE. TEMP(I)) .AND. (TMPR .LE. TEMP(I+1)))
              .  .  .  .  IT1 = I
              .  .  .  .  IT2 = I+1
              .  .  .  .  GO TO 136
              .  .  .  ...FIN
              .  .  ...FIN
      136     .  .  CONTINUE
              .  .  XT(1,1) = ALOG10(T(IT1,ID1))
              .  .  XT(2,1) = ALOG10(T(IT2,ID1))
              .  .  XT(1,2) = ALOG10(T(IT1,ID2))
              .  .  XT(2,2) = ALOG10(T(IT2,ID2))
              .  .  XNT = ALOG10(TMPR/TEMP(IT1))/ALOG10(TEMP(IT2)/TEMP(IT1))
              .  .  XCT(1) = XT(1,1) + XNT*(XT(2,1) - XT(1,1))
              .  .  XCT(2) = XT(1,2) + XNT*(XT(2,2) - XT(1,2))
              .  .  CFT = XCT(1) + (XCT(2) - XCT(1))*XD/XDEN
              .  .  CFT = 10.**CFT
              .  ...FIN
      C       .
      C       .  *** FINE SEDIMENT LOAD CORRECTION ***
      C       .
              .  WHEN (FSL .LE. 10.) CFF=1.0
              .  ELSE
              .  .  ID1 = 0
              .  .  ID2 = 0
              .  .  DO (I=1,9)
              .  .  .  IF((FHRAD .GE. DF(I)) .AND. FHRAD .LE. DF(I+1))
              .  .  .  .  ID1 = I
              .  .  .  .  ID2 = I+1
              .  .  .  .  GO TO 142
```

```
         .  .  .  ...FIN
         .  .  ...FIN
  142    .  .  CONTINUE
         .  .  WHEN (FSL .GT. 1.E+5)
         .  .  .  WRITE(6,5)
  5      .  .  .  FORMAT(//10X,'***** SUBROUTINE COLBY -- FSL WENT > 1.E+5')
         .  .  .  IF1 = 4
         .  .  .  IF2 = 5
         .  .  ...FIN
         .  .  ELSE
         .  .  .  IF1 = 0
         .  .  .  IF2 = 0
         .  .  .  DO (I=1,4)
         .  .  .  .  IF ((FSL .GE. CF(I)) .AND. (FSL .LE. CF(I+1)))
         .  .  .  .  .  IF1 = I
         .  .  .  .  .  IF2 = I+1
         .  .  .  .  .  GO TO 148
         .  .  .  .  ...FIN
         .  .  .  ...FIN
  148    .  .  .  CONTINUE
         .  .  ...FIN
         .  .  XF(1,1) = ALOG10(F(IF1,ID1))
         .  .  XF(2,2) = ALOG10(F(IF2,ID2))
         .  .  XF(1,2) = ALOG10(F(IF1,ID2))
         .  .  XF(2,1) = ALOG10(F(ID2,ID1))
         .  .  XNT = (FSL - CF(IF1))/(CF(IF2) - CF(IF1))
         .  .  XCT(1) = XF(1,1) + XNT*(XF(2,1) - XF(1,1))
         .  .  XCT(2) = XF(1,2) + XNT*(XF(2,2) - XF(1,2))
         .  .  XNT = ALOG10(FHRAD/DF(ID1))/ALOG10(DF(ID2)/DF(ID1))
         .  .  CFF = XCT(1) + XNT*(XCT(2) - XCT(1))
         .  .  CFF = 10.**CFF
         .  ...FIN
         .  TCF = CFT * CFF - 1.0
         .  CFD = 1.
         .  UNLESS ((DB50 .GE. 0.20) .AND. (DB50 .LE. 0.30))
         .  .  IP1 = 0
         .  .  IP2 = 0
         .  .  DO (I=1,10)
         .  .  .  IF ((DB50 .GE. DP(I)) .AND. (DB50 .LE. DP(I+1)))
         .  .  .  .  IP1 = I
         .  .  .  .  IP2 = I+1
         .  .  .  .  GO TO 153
         .  .  .  ...FIN
         .  .  ...FIN
  153    .  .  CONTINUE
         .  .  P2 = ALOG10(P(IP2))
         .  .  P1 = ALOG10(P(IP1))
         .  .  XNT = ALOG10(DB50/DP(IP1))/ALOG10(DP(IP2)/DP(IP1))
         .  .  CFD = P1 + XNT * (P2-P1)
         .  .  CFD = 10.**CFD
         .  ...FIN
         .  FFF = CFD * TCF
         .  FFF = FFF + 1.0
         .  GSI = FFF * GTUC
  C      .
  C      .  *** CONVERTING GSI FROM (TONS/DAY/FT) TO (KG/DAY/M) ***
```

```
        . GSI = GSI * 2.976328E+3
        ...FIN
        RETURN
        END
```

(FLECS VERSION 22.46)

```
        ----------------------------------------
          SUBROUTINE COLLAP(ALEN,AREA,C,DELZ,EL,IELM,K,NELEM,
         1                 CELCTR,TMASS,AWID,VSET,DFZ,CNODE,XSAREA)
C         INTERPRETS INITIAL CONDITIONS
C             CROSS-SECTION
C             SEDIMENT CONCENTRATIONS KG/M3
C             PARTICULATE CONCENTRATIONS PC/KG
C             DISSOLVED CONCENTRATIONS PC/M3
C         ARRIVES AT BOTH
C             NODAL CONCENTRATIONS
C             ELEMENT AVERAGE VALUES
C         NOTE (1.-DELZ*WS/EZ) MUST BE > 0
C
          REAL MSAB,MSBB,MSAT,MSBT
C
          INCLUDE 'SY:ELMSIZ.PRM'
C
          DIMENSION AREA(MXELEM),C(MXELEM,MAXCON),IELM(MXELEM),
         1         EL(MXELEM),CELCTR(MXELEM,MAXCON),TMASS(MAXCON),
         2         AWID(MXELEM),VSET(3),DFZ(4),CNODE(MXELEM,MAXCON),
         3         XSAREA(MXELEM)
C
         BLEN=1./ALEN
         ELTOP=DELZ
         WI=AREA(1)*BLEN
         WJ=AREA(2)*BLEN
         DELEV=EL(2)-EL(1)
         CI=C(1,K)
         CJ=C(2,K)
         NELMBT=1
         MSAB=DELEV*(CJ*WJ/3.+CJ*WI/6.+CI*WJ/6.+CI*WI/3.)
         MSBB=0.
         TMASS(K)=0.0
C
         IF(K .NE. 7)
         .  GI=C(1,K+3)
         .  GJ=C(2,K+3)
         .  CON=1./12.
         .  PMSAB=DELEV*(.25*(WJ*CJ*GJ+WI*CI*GI)
         1.        +CON*(WI*CJ*GJ+WJ*CI*GJ+WJ*CJ*GI
         2.              WI*CI*GJ+WI*CJ*GI+WJ*CI*GI))
         .  PMSBB=0.
         .  TMASS(K+3)=0.0
         ...FIN
         DO(I=1,NELEM)
         .  NELMTP=IELM(I)
         .  ELTP=EL(NELMTP+1)
         .  ELBT=EL(NELMTP)
         .  WI=AREA(NELMTP)*BLEN
         .  WJ=AREA(NELMTP+1)*BLEN
         .  CI=C(NELMTP,K)
         .  CJ=C(NELMTP+1,K)
         .  FAC1=(ELTOP-ELBT)/(ELTP-ELBT)
         .  FAC2=(ELTP-ELTOP)/(ELTP-ELBT)
```

```
             .  CTOP=CJ*FAC1+CI*FAC2
             .  WTOP=WJ*FAC1+WI*FAC2
             .  MSAT=(ELTP-ELTOP)*(CJ*WJ/3.+CJ*WTOP/6.+CTOP*WJ/6.
            1.       +CTOP*WTOP/3.)
             .  MSBT=(ELTOP-ELBT)*(CTOP*WTOP/3.+CTOP*WI/6.
            1.       +CI*WTOP/6.+CI*WI/3.)
             .  IF(K .NE. 7)
             .  .  GI=C(NELMTP,K+3)
             .  .  GJ=C(NELMTP+1,K+3)
             .  .  GTOP=GJ*FAC1+GI*FAC2
             .  .  PMSAT=(ELTP-ELTOP)*(.25*(WJ*CJ*GJ+WTOP*CTOP*GTOP)
            1.  .        +CON*(WTOP*CJ*GJ+WJ*CTOP*GJ+WJ*CJ*GTOP+
            2.  .              WTOP*CTOP*GJ+WTOP*CJ*GTOP+WJ*CTOP*GTOP))
             .  .  PMSBT=(ELTOP-ELBT)*(.25*(WTOP*CTOP*GTOP+WI*CI*GI)
            1.  .        +CON*(WI*CTOP*GTOP+WTOP*CI*GTOP+WTOP*CTOP*GI
            2.  .        +WTOP*CI*GI+WI*CTOP*GI+WI*CI*GTOP))
             .  ...FIN
             .  INDIC=NELMTP-NELMBT
             .  CMASS=0.
             .  PMASS=0.
             .  IF(INDIC .EQ. 0)
             .  .  CMASS=MSBT-MSBB
             .  .  IF(K, NE. 7) PMASS=PMSBT-PMSBB
             .  ...FIN
             .  IF(INDIC .GE. 1)
             .  .  CMASS=MSBT+MSAB
             .  .  IF(K .NE. 7) PMASS=PMSBT+PMSAB
             .  ...FIN
             .  IF(INDIC .GE. 2)
             .  .  DO(J=NELMBT+1, NELMTP-1)
             .  .  .  CI=C(J,K)
             .  .  .  CJ=C(J+1,K)
             .  .  .  WI=AREA(J)*BLEN
             .  .  .  WJ=AREA(J+1)*BLEN
             .  .  .  CMASS=CMASS+DELEV*(CJ*WJ/3.+CJ*WI/6.+CI*WJ/6.+CI*WI/3.)
             .  .  .  IF(K .NE. 7)
             .  .  .  .  GI=C(J,K+3)
             .  .  .  .  GJ=C(J+1,K+3)
             .  .  .  .  PMASS=PMASS+DELEV*(.25*(WI*CI*GI+WJ*CJ*GJ)+
            1.  .  .  .        CON*(WI*CJ*GJ+WJ*CI*GJ+WJ*CJ*GI+
            2.  .  .  .              WJ*CI*GI+WI*CJ*GI+WI*CI*GJ))
             .  .  .  ...FIN
             .  .  ...FIN
             .  ...FIN
             .  TMASS(K)=TMASS(K)+CMASS*ALEN
             .  IF(K .NE. 7) TMASS(K+3)=TMASS(K+3)+PMASS*ALEN
      C      .
      C      .  DETERMINE CELL CENTERED VALUES
      C      .
             .  CELCTR(I,K)=CMASS/XSAREA(I)
             .  IF(K .NE. 7) CELCTR(I,K+3)=PMASS/XSAREA(I)
      C      .
      C      .  DEVELOP NODAL VALUES OF CONCENTRATION
      C      .
             .  WHEN(I .EQ. 1)
             .  .  WHEN(K .LE. 3)
```

```
C        .   .   .
C        .   .   .   NOTE: CMASS IN KG/M
C        .   .   .
         .   .   .   COEF=0.
         .   .   .   WS=VSET(K)*AREA(I)/(AWID(I)*ALEN)
         .   .   .   EZ=DFZ(K)
         .   .   .   CNODE(1,K)=(2.*CMASS/XSAREA(I)-COEF*DELZ/EZ)/(2.-DELZ*WS/EZ)
         .   .   .   CNODE(2,K)=2.*CMASS/XSAREA(I)-CNODE(1,K)
         .   .   .   CNODE(1,K+3)=(2.*PMASS/XSAREA(I)-COEF*DELZ/EZ)/(2.-WS*DELZ/EZ)
         .   .   .   CNODE(2,K+3)=2.*PMASS/XSAREA(I)-CNODE(1,K+3)
         .   .   ...FIN
         .   .   ELSE
         .   .   .   CNODE(1,K)=CMASS/XSAREA(I)
         .   .   .   CNODE(2,K)=CNODE(1,K)
         .   .   ...FIN
         .   ...FIN
         .   ELSE
         .   .   CNODE(I+1,K)=2.*CMASS/XSAREA(I)-CNODE(I,K)
         .   .   IF(K .LE. 3)CNODE(I+1,K+3)=2.*PMASS/XSAREA(I)-CNODE(I,K+3)
         .   ...FIN
C        .
C        .   OVERWRITE BOTTOM ELEMENTAL NODE INFORMATION
C        .
         .   MSAB=MSAT
         .   MSBB=MSBT
         .   ELTOP=ELTOP+DELZ
         .   NELMBT=NELMTP
         .   IF(K .NE. 7)
         .   .   PMSAB=PMSAT
         .   .   PMSBB=PMSBT
         .   ...FIN
         ...FIN
         RETURN
         END

(FLECS VERSION 22.46)

-------------------------------------
```

```
--------------------------------------
      SUBROUTINE COMB(M, S, Z, R)
C
C     THIS SUBROUTINE MULTIPLIES THE UNSYMMETRIC BAND MATRIX (S)
C     BY THE KNOW  LOAD VECTOR <Z> AND ADDS THE RESULT TO <R>.
C
C     CALLED BY TRANSP.
C
      INCLUDE 'ELMSIZ.PRM'
C
      DIMENSION S(MXELEM,3), R(MXELEM), Y(MXELEM), Z(MXELEM)
C
      R(1)=R(1)+S(1,2)*Z(1)+S(1,3)*Z(2)
      R(M)=R(M)+S(M,1)*Z(M-1)+S(M,2)*Z(M)
      DO(I=2,M-1)
      .  Y(I)=S(I,1)*Z(I-1)+S(I,2)*Z(I)+S(I,3)*Z(I+1)
      .  R(I)=R(I)+Y(I)
      ...FIN
      RETURN
      END

(FLECS VERSION 22,46)
--------------------------------------
```

```
        --------------------------------------

              SUBROUTINE DIAG(ECHO2, ECHO3, ECHO4, ECHO5, ECHO6, ECHO7,
             1              ECHO8, ECHO9, ECHO10, ISEG, JSEG, SAVECH)
              LOGICAL*1 ECHO2, ECHO3, ECHO4, ECHO5, ECHO6, ECHO7, ECHO8,
             1         ECHO9, ECHO10, SAVECH, WRTSEG
              DIMENSION JSEG(5), SAVECH(10)
              ECHO2=.FALSE.
              ECHO3=.FALSE.
              ECHO4=.FALSE.
              ECHO5=.FALSE.
              ECHO6=.FALSE.
              ECHO7=SAVECH(6)
              ECHO8=.FALSE.
              ECHO9=.FALSE.
              ECHO10=.FALSE.
              WRTSEG=.FALSE.
              WHEN(JSEG(1).EQ.0) WRTSEG=.TRUE.
              ELSE
              .  DO (J=1,5)
              .  .  IF(JSEG(J).EQ.ISEG) WRTSEG=.TRUE.
              .  ...FIN
              ...FIN
              IF(WRTSEG)
              .  ECHO2=SAVECH(1)
              .  ECHO3=SAVECH(2)
              .  ECHO4=SAVECH(3)
              .  ECHO5=SAVECH(4)
              .  ECHO6=SAVECH(5)
              .  ECHO8=SAVECH(7)
              .  ECHO9=SAVECH(8)
              .  ECHO10=SAVECH(9)
              ...FIN
              RETURN
              END

        (FLECS VERSION 22.46)

        --------------------------------------
```

```
        ----------------------------------------

              SUBROUTINE DIMDAT(ALEN, AREA, BDIV, BED, DLZSAV, ECHO, ELEV,
             1                   HLDERR, ISEG, NBED, NELEM, NUMERR, PELEV,
             2                   POR, RIVER, XYSO,  EL)
        C
        C    THIS ROUTINE IS RESPONSIBLE FOR READING AND PROCESSING THE DATA
        C    DESCRIBING THE SEGMENT DIMENSIONS AND AREAS
        C
        C      FORMAL PARAMETERS:
        C        ALEN    - SEGMENT LENGTH
        C        AREA    - SEGMENT AREA
        C        BDIV    - STANDARD BED THICKNESS
        C        BED     - INITIAL BED THICKNESS
        C        DLZSAV    - STANDARD ELEMENT THICKNESS
        C        ECHO    - LINE PRINTER ECHO OPTION CONTROL VARIABLE (L*1)
        C      EL      - ELEVATIONS ABOVE THE BED CORRESPONDING TO THE SEGMENT AREAS
        C        ELEV    - ELEVATION OF THE SEGMENT
        C        HLDERR  - HOLDING ARRAY FOR ERROR NUMBERS (BYTE)
        C        ISEG    - CURRENT SEGMENT NUMBER
        C        NBED    - NUMBER OF BED LAYERS
        C        NELEM   - NUMBER OF VERTICAL ELEMENTS
        C        NUMERR  - NUMBER OF INPUT ERRORS DETECTED
        C        PELEV   - UPSTREAM ELEVATION OF SEGMENT NUMBER 1
        C        POR     - PORISITY
        C        RIVER   - SHEAR STRESS COMPUTATION CONTROL VARIABLE
        C        XYSO    - THICKNESS OF THE TOP BED LAYER
        C
        C    CALLED BY: SERATRA
        C    CALLS: PUTERR
        C
              INCLUDE 'ELMSIZ.PRM'
        C
              BYTE HLDERR(100)
        C
              LOGICAL*1 ECHO,RIVER
        C
              DIMENSION AREA(MXELEM), EL(MXELEM)
              RIVER=.TRUE.
        C
        C     CARD 1..........SEGMENT DIMENSIONS
        C
        C       COL.  1- 5...NELEM.....NUMBER OF VERTICAL ELEMENTS
        C             6-10...NBED......NUMBER OF BED LAYERS
        C            11-20...DLZSAV....STANDARD ELEMENT THICKNESS (METERS)
        C            21-30...BDIV......STANDARD BED LAYER THICKNESS
        C            31-40...BED.......INITIAL BED THICKNESS (METERS)
        C            41-50...ALEN......LENGTH OF THE SEGMENT (METERS)
        C            51-60...ELEV......ELEVATION OF THE SEGMENT (METERS)
        C            61-70...POR.......POROSITY
        C            71-80...PELEV.....UPSTREAM ELEVATION OF SEGMENT 1 ONLY
        C                              =0.0:SHEAR STRESS COMPUTED USING VELOCITY
        C                                   DISTRIBUTION AND BED ROUGHNESS
        C                                   (RESERVOIR)
        C                              <>0.0:SHEAR STRESS COMPUTED USING BOTTOM
```

```
C                                            SLOP, HYDRAULIC RADIUS AND SPECIFIC
C                                             WEIGHT OF WATER (FREE FLOWING
C                                             RIVER)
C
            WHEN (ISEG .EQ. 1)
            .  READ(1,1) NELEM,NBED,DLZSAV,BDIV,BED,ALEN,ELEV,POR,PELEV
            .  WHEN (PELEV .EQ. 0.0) RIVER = .FALSE.
            .  ELSE RIVER = .TRUE.
            ...FIN
            ELSE
            .  READ(1,1) NELEM,NBED,DLZSAV,BDIV,BED,ALEN,ELEV,POR
            ...FIN
C
C           *** XYSO -- THICKNESS OF THE TOP BED LAYER
C
            XYSO = BED - (NBED-1) * BDIV
            IF (ECHO)
            .  WRITE(6,5) ISEG
            .  WRITE(6,2) NELEM,NBED,DLZSAV,BDIV,BED,ALEN,ELEV,POR,XYSO
            .  IF (ISEG .EQ. 1)
            .  .  WRITE(6,6) PELEV
            .  .  WHEN (PELEV .EQ. 0.0) WRITE(6,7)
            .  .  ELSE WRITE(6,8)
            .  ...FIN
            ...FIN
C
            IF (NELEM .LT. 0 .OR. NELEM+1 .GT. MXELEM)
            .  CALL PUTERR(13,NUMERR,HLDERR)
            ...FIN
            IF (NBED .LE. 0) CALL PUTERR(6,NUMERR,HLDERR)
            IF (NBED .GT. MAXLEV) CALL PUTERR(4,NUMERR,HLDERR)
            IF (DLZSAV .LE. 0.0) CALL PUTERR(7,NUMERR,HLDERR)
            IF (BDIV .LE. 0.0) CALL PUTERR(8,NUMERR,HLDERR)
            IF (BED .GT. NBED*BDIV .OR. BED .LE. (NBED-1)*BDIV)
            .  CALL PUTERR(9,NUMERR,HLDERR)
            ...FIN
            IF (ALEN .LE. 0.0) CALL PUTERR(10,NUMERR,HLDERR)
            IF (ELEV .LE. 0.0) CALL PUTERR(11,NUMERR,HLDERR)
            IF (POR .GT. 1.0) CALL PUTERR(12,NUMERR,HLDERR)
C
C           CARD 2..........AREA OF EACH ELEMENT
C
            DO (I=1,MXELEM)
            .  AREA(I)=0.0
            .  EL(I)=0.0
            ...FIN
            READ(1,9) NAREA, VPAREA, DELEV
            WHEN (NAREA .EQ. 0)
            .  ELE=0.
            .  DO(I=1,MXELEM)
            .  .  AREA(I)=VPAREA
            .  .  EL(I)=ELE
            .  .  ELE=ELE+DELEV
            .  ...FIN
            ...FIN
            ELSE
```

```
          .  READ (1,3) (AREA(I), I=1,NAREA)
          .  READ(1,3) (EL(I), I=1,NAREA)
          .  IF (NAREA.LT.MXELEM)
          .  .  DO (I = NAREA+1,MXELEM)
          .  .  .  AREA(I) = VPAREA
          .  .  .  EL(I) = EL(I-1) + DELEV
          .  .  ...FIN
          .  ...FIN
          ...FIN
          IF(ECHO)
          .  WRITE(6,4) (I, AREA(I), I=1,MXELEM)
          .  WRITE(6,10)(I,EL(I),I=1,MXELEM)
          ...FIN
C
          RETURN
C
    1     FORMAT(2I5,7F10.0)
    2     FORMAT(1H0,13X,I5,'...NUMBER OF VERTICAL ELEMENTS'/
         1  14X,I5,'...NUMBER OF BED LAYERS'/
         2  7X,1PE12.5,'...STANDARD ELEMENT THICKNESS (METERS)'/
         3  7X,1PE12.5,'...STANDARD BED LAYER THICKNESS (METERS)'/
         4  7X,1PE12.5,'...INITIAL BED THICKNESS (METERS)'/
         5  7X,1PE12.5,'...LENGTH OF THE SEGMENT (METERS)'/
         6  7X,1PE12.5,'...SEGMENT ELEVATION (METERS)'/
         7  7X,1PE12.5,'...POROSITY'/
         8  7X,1PE12.5,'...THICKNESS OF THE TOP BED LAYER (CALCULATED)')
    3     FORMAT(8F10.0)
    4     FORMAT(1H0,58X,'ELEMENT AREAS'/4(27X,5(I3,1PE12.5)/))
    5     FORMAT(1H0,54X,'INPUT DATA FOR SEGMENT ',I3)
    6     FORMAT(7X,1PE12.5,'...UPSTREAM ELEVATION (METERS)')
    7     FORMAT(19X,'...SHEAR STRESS VALUES COMPUTED USING METHOD'
         1 ' FOR RESERVOIR')
    8     FORMAT(19X,'...SHEAR STRESS VALUES COMPUTED USING METHOD'
         1 ' FOR FREE FLOWING RIVERS')
    9     FORMAT(I5, 7F10.0)
   10     FORMAT(1H0,58X,'NODAL ELEVATIONS'/4(27X,5(I3,1PE12.5)/))
C
          END

(FLECS VERSION 22.46)

--------------------------------------
```

```
      ----------------------------------------
      SUBROUTINE DISOLV(ABAR,  B, BDIV, C, CCIN, COLD, DECAY, DELZ,
     1                  DELTD, DENS, DIAM,  I, KAY1, KAY2,
     2          NELEM, NBED, PCOEF, POR, QHIN, QHOUT, QV, SORBK, ALFA,
     3                 BETA, VEL1, VEL2, BETA1, BETA2,
     4                 DEPO, SCOUR, BEDSD, XYSO, AREA, DSOPB)
C
C    THIS SUBROUTINE CALCULATES COEFFICIENTS OF CONVECTIVE, DECAY
C     AND SOURCE/SINK TERMS OF DISSOLVED POLLUTANT CONVECTION-DIFFUSION
C    EQUATION
C
C        INPUT PARAMETERS:
C           ABAR   - AVERAGE AREA
C        AREA   - VERTICAL PROJECTION AREAS (M2)
C           B      - BED CONCENTRATIONS
C           BDIV   - STANDARD BED LAYER THICKNESS
C           C      - NODAL CONCENTRATION
C           CCIN   - CONCENTRATION OF INFLOW
C           COLD   - CELL-CENTERED CONCENTRATION
C           DECAY  - DECAY VALUES
C           DELTD  - TIME STEP IN DAYS
C           DENS   - DENSITY
C        DEPO   - DEPOSITION RATE (KG(PC)/M2/DAY)
C           DIAM   - PARTICLE DIAMETERS
C           DELZ    - THICKNESS OF THE ELEMENT
C           I      - ELEMENT INDEX
C           KAY1   - LIGHT EXTINCTION COEFFICIENT OF WATER
C           KAY2   - LIGHT EXTINCTION COEFFICIENT OF SUSPENDED
C                    SEDIMENT IN WATER
C           NELEM  - NUMBER OF VERTICAL ELEMENTS
C           PCOEF  - 1ST TERM OF THE PHOTOLYSIS RATE EQUATION, COMPUTED
C                    IN SUBROUTINE PHOINP.
C           POR    - POROSITY
C           QHIN   - INFLOW DISCHARGE
C           QHOUT  - OUTFLOW DISCHARGE
C           QV     - VERTICAL DISCHARGE
C        SCOUR  - SCOUR RATE (KG(PC)/M2/DAY)
C           SORBK  - ADSORBTION ON SEDIMENT
C        DSORB  - DESORPTION FROM SEDIMENT
C        XYSO   - TOP BED LAYER THICKNESS
C        OUTPUT PARAMETERS:
C           ALFA   - DECAY TERM
C        BEDSD  - SCOUR OR DEPOSITION OF ABSORPED CONTAMINANT
C                 (PC/M2/DAY) WHEN NO SCOUR IS TAKING PLACE
C           BETA   - SOURCE OR SINK TERM
C        BETA1  - INFLUENT SOURCE TERM FOR I-TH NODE
C        BETA2  - INFLUENT SOURCE TERM FOR I+1 TH NODE
C           VEL1   - FIRST CONVECTIVE TERM
C           VEL2   - SECOND CONVECTIVE TERM
C
C        CALLED BY TRANSP.
C
      INCLUDE 'ELMSIZ.PRM'
C
```

```
          REAL KAY,KAY1,KAY2
C
          DIMENSION ABAR(MXELEM),  B(MAXLEV,MAXCON-1), BEDSD(3),
         1          CCIN(MXELEM,MAXCON),COLD(MXELEM,MAXCON), DECAY(6),
         2          QHIN(MXELEM), QHOUT(MXELEM), QV(MXELEM), SORBK(9),
         3          DENS(3), DIAM(3), C(MXELEM,MAXCON),CBAR(MXELEM,MAXCON),
         4    DEPO(6), SCOUR(6), AREA(MXELEM), DSORB(9)
C
          DATA ZERO/1.0E-30/
C
C         CONVECTIVE TERM
C
          AQ =QV(I)
          VEL1=AQ/ABAR(I)
          AQ=QV(I+1)
          VEL2=AQ/ABAR(I)
C       DECAY TERM
          TOTDK = 0.0
          WHEN (PCOEF .NE. 0.0) COMPUTE-PHOTOLYSIS-RATE-FOR-ELEMENT-I
          ELSE PHOTO = 0.0
          DO (IJI=1,5) TOTDK = TOTDK + DECAY(IJI)
          ALFA=QHOUT(I)/(ABAR(I)*DELZ)+TOTDK+PHOTO
          IF(I .EQ. NELEM) ALFA = ALFA + DECAY(6)
C
C       SOURCE OR SINK TERM
          BETA1=QHIN(I)/(ABAR(I)*DELZ)*(CCIN(I,7)/3.+CCIN(I+1,7)/6.)
          BETA2=QHIN(I)/(ABAR(I)*DELZ)*(CCIN(I,7)/6.+CCIN(I+1,7)/3.)
C *************************************************************************
C *                                                                       *
C *   WARNING: THE VALUE OF CBAR SHOULD BE UPDATED BY ITERATIVELY         *
C *            SOLVING FOR C AT THE ADVANCED TIME, AND APPROXIMATING      *
C *            CBAR AS THE NEW AVERAGE CONCENTRATION OVER THE TIME        *
C *            STEP.                                                      *
C *                                                                       *
C *************************************************************************
          DO (IE = 1,MXELEM)
          .  DO (IC = 1,MAXCON)
          .  .  CBAR(IE,IC) = CCIN(IE,IC)
          .  ...FIN
          ...FIN
          DO (J = 1,3)
          .  JP3 = J + 3
          .  IP1 = I + 1
C         .
          .  IF(CBAR(I,J).GT.0.0.AND.CBAR(IP1,J).GT.0.0)
          .  .  ADDS1 =        (SORBK(J)*SORBK(JP3)/12.*(3.*CBAR(I,J)*CBAR(I,7)
         1.  .                 +CBAR(I,J)*CBAR(IP1,7)+CBAR(IP1,J)*CBAR(I,7)+
         2.  .                 CBAR(IP1,J)*CBAR(IP1,7)) - SORBK(JP3)/6.*(2.*
         3.  .                 CBAR(I,JP3) + CBAR(IP1,JP3)))
          .  .  DSAD1 =        (DSORB(J)*DSORB(JP3)/12.*(3.*CBAR(I,J)*CBAR(I,7)
         1.  .                 +CBAR(I,J)*CBAR(IP1,7)+CBAR(IP1,J)*CBAR(I,7)+
         2.  .                 CBAR(IP1,J)*CBAR(IP1,7)) - DSORB(JP3)/6.*(2.*
         3.  .                 CBAR(I,JP3) + CBAR(IP1,JP3)))
          .  .  ADDS2 =       (SORBK(J)*SORBK(JP3)/12.*(CBAR(I,J)*CBAR(I,7)+
         1.  .                CBAR(I,J)*CBAR(IP1,7)+CBAR(IP1,J)*CBAR(I,7)+
         2.  .                3.*CBAR(IP1,J)*CBAR(IP1,7)) - SORBK(JP3)/6.*
```

```
         3.  .                 (CBAR(I,JP3) + 2.*CBAR(I+1,JP3)))
         .   .  DSAD2 =        (DSORB(J)*DSORB(JP3)/12.*(CBAR(I,J)*CBAR(1,7)+
         1.  .                 CBAR(I,J)*CBAR(IP1,7)+CBAR(IP1,J)*CBAR(I,7)+
         2.  .                 3.*CBAR(IP1,J)*CBAR(IP1,7)) - DSORB(JP3)/6.*
         3.  .                 (CBAR(I,JP3) + 2.*CBAR(I+1,JP3)))
         .   .  IF(ADDS1.GT.0.0.OR.ADDS1.EQ.DSAD1)BETA1=BETA1-ADDS1
         .   .  IF(DSAD1.LT.0.0)BETA1=BETA1-DSAD1
         .   .  IF(ADDS2.GT.0.0.OR.ADDS2.EQ.DSAD2)BETA2=BETA2-ADDS2
         .   .  IF(DSAD2.LT.0.0)BETA2=BETA2-DSAD2
         .   ...FIN
C        .
         ...FIN
C***********************************************************************
C        TRANSFER BETWEEN DISSOLVED STREAM CONTAMINANT AND ABSORPED    *
C        BED CONTAMINANT IS INCLUDED WHENEVER NO SCOURING OCCURS FOR   *
C        A PARTICULAR SEDIMENT SIZE (EG SAND, SILT, OR CLAY)           *
C***********************************************************************
         BETA = 0.0
         IF (I.EQ.1)
         .  DO (J=1,3)  BEDSD(J) = 0.0
         .  IF(NBED.GT.0)
         .  .  DO (J=1,3)
         .  .  .  WHEN(SCOUR(J).GT.0.0.OR.B(NBED,J).LE.ZERO)
         .  .  .  .  BEDSD(J)=0.0
         .  .  .  ...FIN
         .  .  .  ELSE
         .  .  .  .  RHOJ=B(NBED,J)*(1.0-POR)*DENS(J)
         .  .  .  .  D = DIAM(J)
         .  .  .  .  IF(D.GT.XYSO) D=XYSO
         .  .  .  .  RATE = SORBK(J+6)*(SORBK(J)*(CBAR(1,7)+CBAR(2,7))/2.
         1.  .  .  .           -B(NBED,J+3))* D * RHOJ
         .  .  .  .  BETA = BETA - RATE /DELZ
         .  .  .  .  BEDSD(J)=-RATE
         .  .  .  ...FIN
         .  .  ...FIN
         .  ...FIN
         ...FIN
         RETURN

-----------------------------------------

         TO COMPUTE-PHOTOLYSIS-RATE-FOR-ELEMENT-I
         .  AVGSED = 0.0
         .  WHEN (I .EQ. NELEM)
         .  .  DO (IJI=1,3) AVGSED = AVGSED + C(I+1,IJI)
         .  ...FIN
         .  ELSE
         .  .  DO (IK=I+1,NELEM+1)
         .  .  .  DO (IJI=1,3) AVGSED = AVGSED + C(IK,IJI)
         .  .  ...FIN
         .  .  AVGSED = AVGSED / (NELEM+1-I)
         .  ...FIN
         .  KAY = KAY1 + KAY2 * AVGSED
         .  WHEN (I .EQ. NELEM) TERM1 = 1.0
         .  ELSE TERM1 = EXP (-KAY*(NELEM-I)*DELZ)
         .  TERM2 = (1.0 - EXP(-KAY*DELZ)) / (KAY*DELZ)
```

```
(FLECS VERSION 22.46)      28-JAN-82   10:23:06   PAGE  00004

 00163          .  PHOTO = PCOEF * TERM1 * TERM2
 00164          ...FIN
 00165          END

       ----------------------------------------

             PROCEDURE CROSS-REFERENCE TABLE

       00148 COMPUTE-PHOTOLYSIS-RATE-FOR-ELEMENT-I
             00072


       (FLECS VERSION 22.46)

       ----------------------------------------
```

```
        ----------------------------------------
        SUBROUTINE EQUPCS(PXSAR, UWID, UEL, XSAREA, NELEM, MELEM,
       1             RATIO, IELP, HEQXS, FERROR, DELTA)
C
C       THIS SUBROUTINE FINDS CROSS-SECTIONAL AREAS AND HEIGHTS WITHIN THE
C       UPSTREAM CROSS-SECTION WHICH CORRESPONDS TO THE SEGMENT IMMEDIATELY
C       DOWNSTREAM.
C
        INCLUDE 'ELMSIZ.PRM'
        LOGICAL*1 FERROR
C
        DIMENSION PXSAR(MXELEM), UWID(MXELEM), UEL(MXELEM),
       1          XSAREA(MXELEM), IELP(MXELEM), HEQXS(MXELEM)
C
        FERROR=.FALSE.
        IP=1
        PXS=PXSAR(1)
        UBTM=UWID(1)
        ELBTM=UEL(1)
        TEMPXS=0.
C
        DO(I=1,NELEM)
        .  XS=RATIO*XSAREA(I)
        .  UNTIL(XS .LE. PXS .OR. IP .EQ. MELEM)
        .  .  IP=IP+1
        .  .  TEMPXS=PXS
        .  .  UBTM=UWID(IP)
        .  .  ELBTM=UEL(IP)
        .  .  PXS=PXS+PXSAR(IP)
        .  ...FIN
        .  IELP(I)=IP
        .  WHEN(XS .EQ. PXS)
        .  .  HEQXS(I)=UEL(IP+1)
        .  .  IF(I.EQ.NELEM) HEQXS(I)= UEL(IP) + DELTA
        .  .  ELBTM=UEL(IP+1)
        .  .  UBTM=UWID(IP+1)
        .  .  PXS=0
        .  .  TEMPXS=0
        .  ...FIN
        .  ELSE
        .  .  A=(UWID(IP+1)-UBTM)/(2.*(UEL(IP+1)-ELBTM))
        .  .  B=UBTM
        .  .  C=XS-TEMPXS
        .  .  WHEN (A .EQ. 0.) HEQXS(I)=C/B+ELBTM
        .  .  ELSE
        .  .  .  BSQ4AC=B*B+4.*A*C
        .  .  .  IF (BSQ4AC .LT. 0.) GO TO 200
        .  .  .  HEQXS(I)=(SQRT(BSQ4AC)-B)/2./A+ELBTM
        .  .  ...FIN
        .  .  UBTM=2.*A*(HEQXS(I)-ELBTM)+B
        .  .  ELBTM=HEQXS(I)
        .  .  PXS=PXS-XS
        .  .  TEMPXS=0.
        .  ...FIN
```

```
        ...FIN
        RETURN
  200   CONTINUE
        FERROR=.TRUE.
        WRITE(6,1)
  1     FORMAT(10X,'FATAL ERROR - BS04AC IN EQUPCS SUBROUTINE IS < 0')
        RETURN
        END
```

(FLECS VERSION 22.46)

```
(FLECS VERSION 22.46)      13-MAR-81    13:24:30  PAGE  00001

        -------------------------------------

              SUBROUTINE EQUPXS(PXSAR, PWID,PDELZ,XSAREA,NELEM,MELEM,
             1             RATIO, IELP, HEQXS)
        C
              INCLUDE 'SY:ELMSIZ.PRM'
        C
              DIMENSION PXSAR(MXELEM), PWID(MXELEM), XSAREA(MXELEM),
             1            IELP(MXELEM), HEQXS(MXELEM)
        C
              IP=1
              PXS=PXSAR(1)
              WBTM=PWID(1)
              ELBTM=0.
              TEMPXS=0.
        C
              DO(I=1,NELEM)
              .   XS=RATIO*XSAREA(I)
              .   UNTIL(XS .LE. PXS .OR. IP .EQ. MELEM)
              .   .   IP=IP+1
              .   .   TEMPXS=PXS
              .   .   WBTM=PWID(IP)
              .   .   ELBTM=PDELZ*(IP-1)
              .   .   PXS=PXS+PXSAR(IP)
              .   ...FIN
              .   IELP(I)=IP
              .   WHEN(XS .EQ. PXS)
              .   .   HEQXS(I)=PDELZ * IP
              .   .   PXS=0.
              .   .   TEMPXS=0.
              .   .   ELBTM=IP*PDELZ
              .   .   WBTM=PWID(IP+1)
              .   ...FIN
              .   ELSE
              .   .   HEQXS(I)=(XS-TEMPXS)/WBTM+ELBTM
              .   .   PXS=PXS-XS
              .   .   TEMPXS=0.
              .   .   ELBTM=HEQXS(I)
              .   ...FIN
              ...FIN
              RETURN
              END

        (FLECS VERSION 22.46)

        -------------------------------------
```

```
        -------------------------------------
              SUBROUTINE FCODE(FNAME,BASE,NBRTP,FTYPE,DEV,UIC1,UIC2)
      C
      C    THIS ROUTINE BUILDS A FILE SPECIFICATION INTO THE OUTPUT
      C    PARAMETER FNAME
      C
      C       BASE  - FIRST FIVE CHARATERS OF THE FILE NAME (BYTE ARRAY)
      C       NBRTP - TIME PLANE NUMBER, THIS BECOMES THE LAST 4 CHARACTERS
      C               OF THE 9 CHARACTER FILE NAME (INTEGER)
      C       FTYPE - THESE 3 CHARACTERS BECOME THE EXTENSION (BYTE ARRAY)
      C       DEV   - DEVICE (BYTE ARRAY)
      C       UIC1  - 1ST UIC (BYTE ARRAY)
      C       UIC2  - 2ND UIC (BYTE ARRAY)
      C
      C    CALLED BY: SERATRA
      C
              BYTE FNAME(27),FTYPE(3),DEV(3),UIC1(3),UIC2(3),COLON,LBRAK,
             1     RBRAK,PERIOD,COMMA,BLANK,BASE(5)
      C
              DATA COLON/':'/
              DATA LBRAK/'['/
              DATA RBRAK/']'/
              DATA PERIOD/'.'/
              DATA COMMA/','/
              DATA BLANK/' '/
      C
              ICAR=1
      C
      C       *** DETERMINE IF A DEVICE HAS BEEN SPECIFIED AND IF SO THE NUMBER
      C           CHARACTERS IN THE SPECIFICATION ***
      C
              N=0
              DO (I=1,3)
              .  IF(DEV(I) .NE. BLANK) N=N+1
              ...FIN
              IF (N .NE. 0)
      C       .  *** TRANSFER DEVICE SPECIFICATION ***
              .  DO (I=1,N)
              .  .  FNAME(ICAR)=DEV(I)
              .  .  ICAR=ICAR+1
              .  ...FIN
      C       .  *** INSERT ":" ***
              .  FNAME(ICAR)=COLON
              .  ICAR=ICAR+1
              ...FIN
      C
      C       *** HAVE UIC'S BEEN SPECIFIED ***
              N=0
              DO (I=1,3)
              .  IF(UIC1(I) .NE. BLANK) N=N+1
              ...FIN
              IF (N .NE. 0)
      C       .
      C       .  *** INSERT LEFT BRACKET ***
```

```
        .  FNAME(ICAR)=LBRAK
        .  ICAR=ICAR+1
C       .
C       .  *** TRANSFER 1ST UIC ***
        .  DO (I=1,N)
        .  .  FNAME(ICAR)=UIC1(I)
        .  .  ICAR=ICAR+1
        .  ...FIN
C       .
C       .  *** INSERT COMMA ***
        .  FNAME(ICAR)=COMMA
        .  ICAR=ICAR+1
C       .
C       .  *** TRANSFER 2ND UIC***
        .  DO (I=1,3)
        .  .  IF (UIC2(I) .NE. BLANK)
        .  .  .  FNAME(ICAR)=UIC2(I)
        .  .  .  ICAR=ICAR+1
        .  .  ...FIN
        .  ...FIN
C       .
C       .  *** INSERT RIGHT BRACKET ***
        .  FNAME(ICAR)=RBRAK
        .  ICAR=ICAR+1
        ...FIN
C
C       *** TRANSFER 5 CHARACTER BASE FILE NAME, ASSUME ALL 5 CHARACTER
C           ARE BEING USED ***
        DO (I=1,5)
        .  FNAME(ICAR)=BASE(I)
        .  ICAR=ICAR+1
        ...FIN
C
C       *** CONVERT TIME PLANE NUMBER TO ASCII AND INSERT IT INTO FNAME **
C
        N=NBPTP
        IDIG=N/1000
        FNAME(ICAR)=IDIG+48
        N=N-IDIG*1000
        IDIG=N/100
        FNAME(ICAR+1)=IDIG+48
        N=N-IDIG*100
        IDIG=N/10
        FNAME(ICAR+2)=IDIG+48
        FNAME(ICAR+3)=(N-IDIG*10)+48
        ICAR=ICAR+4
C
C       *** INSERT PERIOD ***
        FNAME(ICAR)=PERIOD
        ICAR=ICAR+1
C
C       *** TRANSFER THE 3 CHARACTER EXTENSION ***
        DO (I=1,3)
        .  FNAME(ICAR)=FTYPE(I)
        .  ICAR=ICAR+1
        ...FIN
```

```
      C
      C       *** INSERT NULL CHARACTER ***
              FNAME(ICAR)=0
              RETURN
              END

        (FLECS VERSION 22,46)

        --------------------------------------
```

```
---------------------------------------
      SUBROUTINE FDCODE(FNAME,BASE,NBRTP,FTYPE,DEV,UIC1,UIC2)
C
C   THIS ROUTINE SPERATES FNAME INTO 6 COMPONENTS
C
C     BASE  - 5 CHARACTER BASE FILE NAME (BYTE ARRAY)
C     NBRTP - TIME PLANE NUMBER THAT IS THE LAST 4 CHARACTERS OF THE
C             9 CHARACTER FILE NAME (INTEGER)
C     FTYPE - FILE EXTENSION (BYTE ARRAY)
C     DEV   - PHICAL DEVICE SPECIFICATION (BYTE ARRAY)
C     UIC1  - 1ST UIC
C     UIC2  - 2ND UIC
C
C   THE OPTIONAL PARAMETERS DEV, UIC1, AND UIC2 WILL BE SET
C   TO BLANKS IF NOT PRESENT IN THE ORIGINAL FILE SPECIFICATION.
C
C   CALLED BY: STRTUP
C
      BYTE FNAME(27),BASE(5),FTYPE(3),DEV(3),UIC1(3),UIC2(3),
     1     LBRAK,RBRAK,COMMA,PERIOD,COLON,BLANK
C
      DATA LBRAK/'['/
      DATA RBRAK/']'/
      DATA COMMA/','/
      DATA PERIOD/'.'/
      DATA COLON/':'/
      DATA BLANK/' '/
C
C     *** FILE SPECIFICATION HAVE FOUR POSSIBLE FORMS ***
C         (1) FILENAME.EXT
C         (2) DEV:FILENAME.EXT
C         (3) [UIC1,UIC2]FILENAME.EXT
C         (4) DEV:[UIC1,UIC2]FILENAME.EXT
C
C     THE FORM CAN BE DETERMINED BY COUNTING THE FOUR SPECIAL
C     CHARACTERS [  ]  .  :
C
      N=0
      DO (I=1,27)
      .  SELECT (FNAME(I))
      .  .  (COLON) N=N+1
      .  .  (LBRAK) N=N+1
      .  .  (RBRAK) N=N+1
      .  .  (PERIOD) N=N+1
      .  ...FIN
      ...FIN
      ICAR=1
      DO (I=1,3)
      .  DEV(I)=BLANK
      .  UIC1(I)=BLANK
      .  UIC2(I)=BLANK
      ...FIN
      SELECT (N)
      .  (1) DECODE-FORM1
```

```
              .  (2) DECODE-FORM2
              .  (3) DECODE-FORM3
              .  (4) DECODE-FORM4
              ...FIN
              RETURN
        C

        --------------------------------------

              TO DECODE-FORM1
              .  DECODE-FILENAME-EXTENSION
              ...FIN
        C

        --------------------------------------

              TO DECODE-FORM2
              .  DECODE-DEVICE
              .  DECODE-FILENAME-EXTENSION
              ...FIN
        C

        --------------------------------------

              TO DECODE-FORM3
              .  DECODE-UIC
              .  DECODE-FILENAME-EXTENSION
              ...FIN
        C

        --------------------------------------

              TO DECODE-FORM4
              .  DECODE-DEVICE
              .  DECODE-UIC
              .  DECODE-FILENAME-EXTENSION
              ...FIN
        C

        --------------------------------------

              TO DECODE-FILENAME-EXTENSION
              .  DO (I=1,5)
              .  .  BASE(I)=FNAME(ICAR)
              .  .  ICAR=ICAR+1
              .  ...FIN
              .  WHEN (FNAME(ICAR) .NE. PERIOD)
              .  .  ICHAR1=FNAME(ICAR)
              .  .  ICHAR2=FNAME(ICAR+1)
              .  .  ICHAR3=FNAME(ICAR+2)
              .  .  ICHAR4=FNAME(ICAR+3)
              .  .  NBRIP=(ICHAR1-48)*1000+(ICHAR2-48)*100+(ICHAR3-48)*10+ICHAR4-48
        C     .  .  *** SKIP OVER PERIOD ***
              .  .  ICAR=ICAR+5
              .  ...FIN
              .  ELSE
```

```
                .  .  ICAR=ICAR+1
                .  .  NBRTP=0
                .  ...FIN
                .  DO (I=1,3)
                .  .  FTYPE(I)=FNAME(ICAR)
                .  .  ICAR=ICAR+1
                .  ...FIN
                ...FIN
       C

       ---------------------------------------

                TO DECODE-DEVICE
                .  I=1
                .  REPEAT WHILE (FNAME(ICAR) .NE. COLON)
                .  .  DEV(I)=FNAME(ICAR)
                .  .  ICAR=ICAR+1
                .  .  I=I+1
                .  ...FIN
       C        .  *** SKIP OVER COLON ***
                .  ICAR=ICAR+1
                ...FIN
       C

       ---------------------------------------

                TO DECODE-UIC
       C        .  *** SKIP OVER LEFT BRACKET ***
                .  ICAR=ICAR+1
                .  I=1
                .  REPEAT WHILE (FNAME(ICAR) .NE. COMMA)
                .  .  UIC1(I)=FNAME(ICAR)
                .  .  ICAR=ICAR+1
                .  .  I=I+1
                .  ...FIN
       C        .  *** SKIP OVER COMMA ***
                .  ICAR=ICAR+1
                .  I=1
                .  REPEAT WHILE (FNAME(ICAR) .NE. RBRAK)
                .  .  UIC2(I)=FNAME(ICAR)
                .  .  ICAR=ICAR+1
                .  .  I=I+1
                .  ...FIN
       C        .  *** SKIP OVER RIGHT BRACKET ***
                .  ICAR=ICAR+1
                ...FIN
                END

       ---------------------------------------

                PROCEDURE CROSS-REFERENCE TABLE

       00060 DECODE-FORM1
             00053

       00064 DECODE-FORM2
```

```
        00054

00115 DECODE-UIC
        00070   00076

00069 DECODE-FORM3
        00055

00074 DECODE-FORM4
        00056

00080 DECODE-FILENAME-EXTENSION
        00061   00066   00071   00077

00104 DECODE-DEVICE
        00065   00075


(FLECS VERSION 22.46)

--------------------------------------
```

```
--------------------------------------

      SUBROUTINE HYDDAT(ALEN, AREA, DELTH, DELZ, D50, ECHO, HLDERR,
     1                  NSETS, NUMERR, SIMLEN, DEPMIN, DLZSAV, EL)
C     DEPMIN HAS BEEN ADDED TO THE SUBROUTINE CALL
C
C    THIS ROUTINE IS RESPONSIBLE FOR READING AND PROCESSING THE
C    HYDROLOGY DATA.  THE DATA IS READ FROM THE INPUT STREAM (LUN 1)
C    AND WRITTEN TO "HYDROLOGY.TMP" (LUN 4) FOR USE DURING THE
C    SIMULATION.
C
C      FORMAL PARAMETERS:
C        ALEN   - SEGMENT LENGTH
C        AREA   - CROSS SECTIONAL AREA OF EACH ELEMENT
C        DELTH  - TIME STEP IN SECONDS
C        DELZ   - STANDARD ELEMENT THICKNESS
C        D50    - MEDIAN BED SEDIMENT DIAMETER
C        ECHO   - LINE PRINTER OPTION CONTROL VARIABLE (L*1)
C        HLDERR - HOLDING ARRAY FOR ERROR NUMBERS (BYTE)
C        NSETS  - NUMBER OF TIMES INITIAL CONDITIONS MUST BE WRITTEN
C                 TO OUTFLO. (NSETS * DELTH) = THE AMOUNT OF TIME IT
C                 TAKES THE FLOW TO PASS THROUGH THE SEGMENT.
C        NUMERR - NUMBER OF INPUT ERRORS
C        SIMLEN - SIMULATION LENGTH - SECONDS (I*4)
C
C    CALLED BY: SERATRA
C    CALLS: PUTERR
C
      INCLUDE 'ELMSIZ.PRM'
C
      BYTE HLDERR(100)
C
      INTEGER*4 ENDTIM,PRETIM,SIMLEN
C
      LOGICAL*1 ECHO
C
      DIMENSION ABAR(MXELEM), AREA(MXELEM), AWID(MXELEM),
     1          EL(MXELEM),XSAREA(MXELEM),BWID(MXELEM),IELM(MXELEM)
C
      REWIND 4
      NSETS=1
      PEND = 0.0
      IDELTH=IFIX(DELTH)
      REPEAT UNTIL (ENDTIM .EQ. -9999)
C     .
C     .  CARD 12.........HYDROLOGY DATA -- THIS DATA IS WRITTEN TO LUN 4
C     .
C     .    COL.  1-10...ENDTIM...ENDING TIME FOR THEDATA ON THE CARD. (SEC)
C     .                           AN ENTRY OF -9999 TERMINATES THE DATA.
C     .         11-20...QI.......TOTAL DISCHARGE OF THIS SEGMENT (M**3/SEC)
C     .         21-30...QO.......TOTAL DISCHARGE OUT OF THIS SEGMENT M**3/SEC)
C     .         31-40...DEPTH....FLOW DEPTH (METERS)
C     .         41-50...TEMPR....WATER TEMPERATURE
C     .
      .  READ(1,1) ENDTIM,QI,QO,DEPTH,TEMPR
```

```
             .  IF (ENDTIM .NE. -9999)
             .  .  IF (MOD(ENDTIM,IDELTH) .NE. 0)
             .  .  .  WRITE(11,10) ENDTIM,IDELTH
      10     .  .  .  FORMAT(' WARNING**** ENDTIM',I10,' IS NOT A MUTIPLE OF DELTH',I10)
             .  .  .  ENDTIM=(ENDTIM/IDELTH+1)*IDELTH
             .  .  ...FIN
             .  ...FIN
    C        .
    C        .
             .  IF (ENDTIM .NE. -9999) PRETIM = ENDTIM
             .  UNLESS (ENDTIM .EQ. -9999)
    C        .  .  MINIMUM DEPTH FLAG TO DIVERT FROM FURTHER CALCULATION
             .  .  NELEM=0
             .  .  VOL=0.
             .  .  VEL=0.
             .  .  DO(I=1,MXELEM)
             .  .  .  AWID(I)=0.
             .  .  .  BWID(I)=0.
             .  .  .  ABAR(I)=0.
             .  .  .  IELM(I)=0.
             .  .  .  XSAREA(I)=0.
             .  .  ...FIN
             .  .  IF (DEPTH .GT. DEPMIN)
    C        .  .  .
    C        .  .  .  *** COMPUTE: NELEM....NUMBER OF ELEMENTS CONTAINED WITHIN DEPTH
    C        .  .  .                ABAR(I)..AVERAGE AREA OF ELEMENTS I AND I+1
    C        .  .  .                AWID(I)..WIDTH OF ELEMENT
    C        .  .  .                VOL......TOTAL VOLUME OF THE SEGMENT
             .  .  .  VOL=0.
             .  .  .  NELEM=DEPTH/DLZSAV
             .  .  .  DELZ=DEPTH/NELEM
             .  .  .  WHEN(NELEM .LE. 1 .OR. NELEM+1 .GT. MXELEM)
             .  .  .  .  CALL PUTERR (14, NUMERR, HLDERR)
             .  .  .  .  WRITE(6,6)
             .  .  .  ...FIN
             .  .  .  ELSE
             .  .  .  .  CALL TRNPOS(ABAR,AREA,AWID,ALEN,BWID,DELZ,EL,IELM,NELEM,
          1. .  .  .  .  XSAREA,VOL)
             .  .  .  .  CALL RADIUS (ALEN, AREA, CROSEC, DEPTH, EL, HRAD)
             .  .  .  ...FIN
    C        .  .  .
    C        .  .  .  IT IS IMPLICITLY ASSUMED THAT A DOWNSTREAM COURANT
    C        .  .  .  NUMBER AT OR NEAR UNITY HAS BEEN EMPLOYED IN THIS
    C        .  .  .  ANALYSIS
    C        .  .  .
             .  .  .  VEL=(QI + QO)/2. * ALEN/VOL
             .  .  ...FIN
             .  .  WRITE(4) ENDTIM,NELEM,DELZ,QI,QO,VOL,VEL,AWID,AREA,TEMPR,
          1. .  .  XSAREA,IELM,DEPTH,BWID,ABAR, HRAD, CROSEC
    C        .  .
             .  .  IF (ECHO)
             .  .  .  WRITE(6,2) ENDTIM,NELEM,TEMPR,QI,QO,DEPTH
             .  .  .  WRITE(6,3)
             .  .  .  DO (I=1,NELEM)
             .  .  .  .  WRITE(6,4)I,AWID(I),ABAR(I),AREA(I),EL(I)
             .  .  .  ...FIN
```

```
              .  .  ...FIN
              .  ...FIN
              ...FIN
              IF (PRETIM .LT. SIMLEN) CALL PUTERR(25, NUMERR, HLDERR)
              REWIND 4
        C
              RETURN
        C
         1    FORMAT(I10,4F10.0)
         2    FORMAT(1H0,58X,'HYDROLOGY DATA'/
             1  9X,I10,'...DATA SET ENDING TIME'/
             2  14X,I5,'...NUMBER OF ELEMENTS WITHIN THE FLOW DEPTH'/
             3  7X,1PE12.5,'...WATER TEMPERATURE'/
             6  7X,1PE12.5,'...TOTAL DISCHARGE OF THIS SEGMENT'/
             7  7X,1PE12.5,'...TOTAL DISCHARGE OUT OF THIS SEGMENT'/
             8  7X,1PE12.5,'...FLOW DEPTH')
         3    FORMAT(1H0,'ELEMENT',4X,'SEGMENT',14X,'AVERAGE',12X,'NODE VP-AREA',
             1  11X,'NODE ELEV'/1X,'NUMBER',6X,'WIDTH',12X,'ELEMENT AREA')
         4    FORMAT(3X,I2,4X,1PE12.5,9X,1PE12.5,10X,1PE12.5,10X,1PE12.4)
         6    FORMAT(//50X,'DEPTH TOO GREAT FOR THE MAXIMUM NUMBER OF'
             1                 ' ELEMENTS'//)
        C
              END

        (FLECS VERSION 22.46)

        ----------------------------------------
```

```
      ----------------------------------------
          SUBROUTINE HYDFLO(ALEN, AREA, AWID, DELZ,  DEPTH, D50,
        1                    ELEV, ENDHYD, ETIME, FERROR, HRAD, NELEM,
        2                    NEWQI, PELEV, QHIN, QHOUT, QV, RIVER, SLOPE,
        3                    STRESS, TEMPR, VEL, VOL, DEPMIN,
        4                    XSAREA, BWID, ABAR, QI, CROSEC)
C
C    THIS SUBROUTINE IS CALLED EACH TIME STEP  TO READ ANY NEW HYDROLOGY
C    DATA THAT WAS WRITTEN TO LUN 4 BY SUBROUTINE HYDDAT.
C
C      FORMAL PARAMETERS:
C        ALEN   - LENGTH OF THE SEGMENT
C        AREA   - AREA OF EACH ELEMENT
C        AWID   - ELEMENT WIDTHS
C        DELZ   - STANDARD ELEMENT THICKNESS
C        DEPTH  - FLOW DEPTH
C        D50    - MEDIAN BED SEDIMENT DIAMETER (METER)
C        ELEV   - SEGMENT ELEVATION
C        ENDHYD - ENDING TIME OF THE CURRENT HYDROLOGY DATA (I*4)
C        ETIME  - ELAPSED TIME OF THE SIMULATION (I*4)
C        FERROR - FATAL ERROR FLAG (L*1)
C        HRAD   - HYDRAULIC RADIUS
C        NELEM  - NUMBER OF ELEMENTS
C        NEWQI  - NEW QI DATA FLAG (L*1)
C        PELEV  - ELEVATION OF THE UPSTREAM SEGMENT
C        QHIN   - INFLOW DISCHARGE
C        QHOUT  - OUTFLOW DISCHARGE
C        QV     - VERTICAL FLOWS
C        RIVER  - SHEAR STREES COMPUTATION CONROL VARIABLE (L*1)
C        SLOPE  - BED SLOPE
C        STRESS - BED SHEAR STRESS
C        TEMPR  - WATER TEMPERATURE
C        VEL    - FLOW VELOCITY OF QHOUT
C        VOL    - SEGMENT VOLUME
C
C    CALLED BY: SERATRA
C    CALLS: SHEARR, SHEARS, PROFIL
C
          INCLUDE 'ELMSIZ.PRM'
C
          INTEGER*4 ETIME,ENDHYD
C
          LOGICAL*1 NEWQI,RIVER,FERROR
C
          DIMENSION AREA(MXELEM), AWID(MXELEM), QHIN(MXELEM),
        1           QHOUT(MXELEM), QV(MXELEM),
        2          XSAREA(MXELEM), BWID(MXELEM), ABAR(MXELEM),
        3          IELM(MXELEM),DUMMY(MXELEM)
C
          DATA SECDAY/86400./
          DATA RHO/1000./
          DATA ZERO/1.0E-05/
C
C
```

```
          FERROR = .FALSE.
          NEWQI = .FALSE.
          IF (ETIME .GT. ENDHYD)
          .  NEWQI = .TRUE.
          .  REPEAT UNTIL (ETIME .LE. ENDHYD)
          .  .  READ(4,END=200) ENDHYD,NELEM,DELZ,QI,QO,VOL,VEL,AWID,AREA,TEMPR,
         1.  .                 XSAREA,IELM,DEPTH,BWID,ABAR,HRAD,CROSEC
          .  ...FIN
          .    IF (DEPTH .GE. DEPMIN)
          .  .  WHEN (RIVER)
          .  .  .  CALL SHEARS(ALEN,ELEV,HRAD,PELEV,SLOPE,STRESS,USTAR)
          .  .  ...FIN
          .  .  ELSE
          .  .  .  CALL SHEARR(DEPTH,D50,STRESS,USTAR,VEL)
          .  .  .  SLOPE=STRESS/(RHO*HRAD)
          .  .  ...FIN
          .  .  CALL PROFIL(ALEN, AWID, DELZ,  DEPTH, NELEM, QI, USTAR,
         1.  .              VOL, QHIN,DUMMY,0,DELZ)
          .  .  CALL PROFIL(ALEN, AWID, DELZ,  DEPTH, NELEM, QO, USTAR,
         1.  .              VOL, QHOUT,DUMMY,0,DELZ)
C         .  .  *** CONVERT UNITS TO M**3/DAY ***
          .  .  DO (J=1,NELEM)
          .  .  .  QHIN(J) = QHIN(J) * SECDAY
          .  .  .  QHOUT(J) = QHOUT(J) * SECDAY
          .  .  ...FIN
C         .  .
C         .  .  *** COMPUTE VERTICAL FLOWS ***
          .  .  QV(1) = 0.0
          .  .  DO (J=1,NELEM)
          .  .  .  QV(J+1) = QHIN(J) - QHOUT(J) + QV(J)
          .  .  ...FIN
          .  ...FIN
          ...FIN
          RETURN
C
  200     CONTINUE
          FERROR = .TRUE.
          WRITE(6,1)
    1     FORMAT(10X,'FATAL ERROR - HYDROLOGY DATA EXHAUSTED')
          RETURN
  300     CONTINUE
          FERROR=.TRUE.
          WRITE(6,3)
          WRITE(6,2) (J,QV(J),J=1,NELP1)
    3     FORMAT(10X,'FATAL ERROR - VERTICAL FLUX COMPUTATION')
    2     FORMAT(15X,I5,1PE12.4)
          RETURN
C
          END
```

(FLECS VERSION 22.46)

```
          ------------------------------------
        SUBROUTINE ICFLO(CCIN,DEPTH,DELZ,D50,ENDIC,ETIME,FERROR,
     1                    INFLO, ISEG, NELEM,  NEWIC,
     2                    QHIN,QI,DEPMIN,ALEN,
     3     UEL, UWID, XSAREA, AREA, AWID, DFZ, VSET,
     4                    EL,ELEV,PELEV,RIVER,NEWQI,NEWTRB)
C
C
C
C       THIS ROUTINE IS CALLED EACH TIME STEP  TO READ THE INITIAL
C       CONDITIONS TO THE FIRST SEGMENT OR THE INFLOWS
C       FROM THE PREVIOUS SEGMENT.
C
C          FORMAL PARAMETERS:
C             CCIN   - CONCENTRATION OF INFLOWS- CELL CENTERED
C             DEPTH  - FLOW DEPTH OF THE CURRENT SEGMENT
C             DELZ   - STANDARD ELEMENT THICKNESS OF THE CURRENT SEGMENT
C             ENDIC  - ENDING TIME OF THE CURRENT INITIAL CONDITIONS DATA
C             ETIME  - ELAPSED TIME OF THE SIMULATION (I*4)
C             FERROR - FATAL ERROR FLAG (L*1)
C             INFLO  - LOGICAL UNIT NUMBER FOR DATA FROM PREVIOUS SEGMENT
C             ISEG   - CURRENT SEGMENT NUMBER
C             NELEM  - NUMBER OF ELEMENTS IN THE CURRENT SEGMENT
C             NELEMB - NUMBER OF ELEMENTS IN THE PREVIOUS SEGMENT
C             NEWIC  - INITIAL CONDITIONS FLAG (L*1)
C             PDELZ  - STANDARD ELEMENT THICKNESS OF THE PREVIOUS SEGMENT
C             PDEPTH - FLOW DEPTH OF THE PREVIOUS SEGMENT
C             QHIN   - INFLOW DISCHARGE
C             QHOLD  - DISCHARGE INTO THE SEGMENT FROM THE PREVIOUS ONE
C
C          CALLED BY: SERATRA
C          CALLS: EQUPCS, EQUPXS, PROFIL, RADIUS
C
        INCLUDE 'ELMSIZ.PRM'
C
        INTEGER*4 ENDIC,ETIME
C
        LOGICAL*1 NEWIC,NEWQI,NEWTRB,FERROR,RIVER
C
        DIMENSION CCIN(MXELEM,MAXCON), CNODE(MXELEM,MAXCON),
     1           QHIN(MXELEM), QHOLD(MXELEM),XSAREA(MXELEM),
     2     PXSAR(MXELEM),IELP(MXELEM),UWID(MXELEM),UWDAVG(MXELEM),
     3      HEWXS(MXELEM), UEL(MXELEM),TMASS(MAXCON),PWID(MXELEM),
     4           OCNODE(MXELEM,MAXCON),
     5           AREA(MXELEM), AWID(MXELEM), DFZ(4), VSET(3),
     6      EL(MXELEM), UAREA(MXELEM)
C
        DATA RHO/1000./
        DATA SECDAY/86400./
C
        FERROR = .FALSE.
        NEWIC = .FALSE.
        IDIM=MXELEM
        JDIM=MAXCON
```

```
           WHEN (ISEG .EQ. 1)
 C         .  DISTRIBUTES INITIAL CONDITIONS UPSTREAM OF INITIAL SEGMENT BY
 C         .  CONSERVING RELATIVE CROSS-SECTIONAL AREAS AND DISCHARGES---
 C         .  ASSUMES LINEAR PROFILE OF BOTH SEDIMENT AND PARTICULATE, AND
 C         .  WIDTH A LINEAR FUNCTION OF DEPTH.
 C         .  *** INITIAL CONDITIONS ***
 C         .
           .  IF (ETIME .GT. ENDIC) NEWIC = .TRUE.
           .  IF(NEWIC.OR.NEWQI.OR.NEWTRB)
           .  .  UNTIL (ETIME .LE. ENDIC)
 C         .  .  .
           .  .  .  READ(2,END=200)ENDIC,NM,PDEPTH,((CNODE(I,J),J=1,MAXCON),I=1,NM)
           .  .  ...FIN
           .  .  IF(DEPTH .LE. DEPMIN) RETURN
           .  .  XS=0.
           .  .  PXS=0.
           .  .  CON=1./12.
           .  .  DO(I=1,NM)
           .  .  .  IF(PDEPTH .LE. UEL(I)) GO TO 10
           .  .  .  WT=UWID(I+1)
           .  .  .  WB=UWID(I)
           .  .  .  ET =UEL(I+1)
           .  .  .  EB=UEL(I)
           .  .  .  IF (PDEPTH .LT. ET)
           .  .  .  .  WT=WB+(WT-WB)*(PDEPTH-EB)/(ET-EB)
           .  .  .  .  ET=PDEPTH
           .  .  .  ...FIN
           .  .  .  DELTA=ET-EB
           .  .  .  PXSAR(I)=(WT+WB)*DELTA/2.
           .  .  .  PXS=PXS+PXSAR(I)
           .  .  .  MELEM=I
           .  .  ...FIN
    10     .  .  CONTINUE
           .  .  DO (I=1,NELEM) XS=XS+XSAREA(I)
           .  .  RATIO=PXS/XS
           .  .  CALL EQUPCS(PXSAR,UWID,UEL,XSAREA,NELEM,MELEM,RATIO,IELP,
         1.   .             HEQXS,FERROR, DELTA)
           .  .  IF(FERROR) RETURN
           .  .  DO(I=1,MELEM-1) UWDAVG(I)=(UWID(I)+UWID(I+1))/2.
           .  .  UWDAVG(MELEM)=(UWID(MELEM)+WT)/2.
           .  .  UDEL=UEL(2)
           .  .  UVOL=PXS*ALEN
           .  .  VEL=QI/PXS
 C         .  .
           .  .  DO (I =1,MXELEM) UAREA(I)=UWID(I)*ALEN
           .  .  CALL RADIUS (ALEN,UAREA,CROSEC,PDEPTH,UEL,HRAD)
           .  .  WHEN(RIVER)
           .  .  .  CALL SHEARS(ALEN,ELEV,HRAD,PELEV,SLOPE,STRESS,USTAR)
           .  .  ...FIN
           .  .  ELSE
 C         .  .  .
           .  .  .  CALL SHEARR(PDEPTH,D50,STRESS,USTAR,VEL)
           .  .  .  SLOPE=STRESS/(RHO*HRAD)
           .  .  ...FIN
           .  .  CALL PROFIL(ALEN,UWDAVG,UDEL,PDEPTH,MELEM,QI,USTAR,UVOL,QHOLD,
         1.   .             UEL,1,DELTA)
```

```
        .  .  DO (I=1,MELEM-1)
        .  .  .  QHOLD(I)=QHOLD(I)*SECDAY/(UWDAVG(I)*(UEL(I+1)-UEL(I)))
        .  .  ...FIN
        .  .  QHOLD(MELEM)=QHOLD(MELEM)*SECDAY/(UWDAVG(MELEM)*DELTA)
        .  .  DO(K=1,MAXCON)
        .  .  .  IF(K .LE. 3 .OR. K .EQ. 7)
        .  .  .  .  UPSQO=QHOLD(1)
        .  .  .  .  WT=UWID(2)
        .  .  .  .  WB=UWID(1)
        .  .  .  .  DEL=UEL(2)-UEL(1)
        .  .  .  .  CT=CNODE(2,K)
        .  .  .  .  CB=CNODE(1,K)
        .  .  .  .  NELMBT=1
        .  .  .  .  CMSAB=DEL*(CT*WT/3.+CT*WB/6.+CB*WT/6.+CB*WB/3.)*UPSQO
        .  .  .  .  CMSBB=0.
        .  .  .  .  TMASS(K)=0.0
        .  .  .  .  IF (K .NE. 7)
        .  .  .  .  .  GT=CNODE(2,K+3)
        .  .  .  .  .  GB=CNODE(1,K+3)
        .  .  .  .  .  PMSAB=DEL*(0.25*(WT*CT*GT+WB*CB*GB)+
     1. .  .  .  .  .  CON*(WT*CB*GB+WB*CT*GB+WB*CB*GT+
     2. .  .  .  .  .  WB*CT*GT+WT*CB*GT+WT*CT*GB))*UPSQO
        .  .  .  .  .  PMSBB=0.
        .  .  .  .  .  TMASS(K+3)=0.
        .  .  .  .  .  ...FIN
        .  .  .  .  DO(I=1,NELEM)
        .  .  .  .  .  NELMTP=IELP(I)
        .  .  .  .  .  UPSQO=QHOLD(NELMTP)
        .  .  .  .  .  ET=UEL(NELMTP+1)
        .  .  .  .  .  EB=UEL(NELMTP)
        .  .  .  .  .  WT=UWID(NELMTP+1)
        .  .  .  .  .  WB=UWID(NELMTP)
        .  .  .  .  .  CT=CNODE(NELMTP+1,K)
        .  .  .  .  .  CB=CNODE(NELMTP,K)
        .  .  .  .  .  HEL=HEQXS(I)
        .  .  .  .  .  FAC1=(HEL-EB)/(ET-EB)
        .  .  .  .  .  FAC2=(ET-HEL)/(ET-EB)
        .  .  .  .  .  CTOP=CT*FAC1+CB*FAC2
        .  .  .  .  .  WTOP=WT*FAC1+WB*FAC2
        .  .  .  .  .  CMSAT=(ET-HEL)*(CT*WT/3.+CT*WTOP/6.+CTOP*WT/6.+CTOP*WTOP/3.)
     1. .  .  .  .  .  *UPSQO
        .  .  .  .  .  CMSBT=(HEL-EB)*(CTOP*WTOP/3.+CTOP*WB/6.+CB*WTOP/6.+CB*WB/3.)
     1. .  .  .  .  .  *UPSQO
        .  .  .  .  .  IF(K.NE.7)
        .  .  .  .  .  .  GT=CNODE(NELMTP+1,K+3)
        .  .  .  .  .  .  GB=CNODE(NELMTP,K+3)
        .  .  .  .  .  .  GTOP=GT*FAC1+GB*FAC2
        .  .  .  .  .  .  PMSAT=(ET-HEL)*(.25*(WT*CT*GT+WTOP*CTOP*GTOP)
     1. .  .  .  .  .  .  +CON*(WT*CTOP*GTOP+WTOP*CT*GTOP+WTOP*CTOP*GT+
     2. .  .  .  .  .  .  WTOP*CT*GT+WT*CTOP*GT+WT*CT*GTOP))*UPSQO
        .  .  .  .  .  .  PMSBT=(HEL-EB)*(0.25*(WTOP*CTOP*GTOP+WB*CB*GB)
     1. .  .  .  .  .  .  +CON*(WTOP*CB*GB+WB*CTOP*GB+WB*CB*GTOP+
     2. .  .  .  .  .  .  WB*CTOP*GTOP+WTOP*CB*GTOP+WTOP*CTOP*GB))*UPSQO
        .  .  .  .  .  ...FIN
        .  .  .  .  .  INDIC=NELMTP-NELMBT
        .  .  .  .  .  CMASS=0.
```

```
              .  .  .  .  .  PMASS=0.
              .  .  .  .  .  IF(INDIC .EQ. 0)
              .  .  .  .  .  .  CMASS=CMSBT-CMSBB
              .  .  .  .  .  .  IF(K.NE.7) PMASS=PMSBT-PMSBB
              .  .  .  .  .  ...FIN
              .  .  .  .  .  IF(INDIC.GE.1)
              .  .  .  .  .  .  CMASS=CMSBT+CMSAB
              .  .  .  .  .  .  IF(K.NE.7) PMASS=PMSBT+PMSAB
              .  .  .  .  .  ...FIN
              .  .  .  .  .  IF(INDIC.GE.2)
              .  .  .  .  .  .  DO(J=NELMBT+1,NELMTP-1)
              .  .  .  .  .  .  .  DEL=UEL(J+1)-UEL(J)
              .  .  .  .  .  .  .  CT=CNODE(J+1,K)
              .  .  .  .  .  .  .  CB=CNODE(J,K)
              .  .  .  .  .  .  .  UPSQQ=QHOLD(J)
              .  .  .  .  .  .  .  WT=UWID(J+1)
              .  .  .  .  .  .  .  WB=UWID(J)
              .  .  .  .  .  .  .  CMASS=CMASS+DEL*(CT*WT/3.+CT*WB/6.+CB*WT/6.+CB*WB/3.)*UPSQ0
              .  .  .  .  .  .  .  IF(K.NE.7)
              .  .  .  .  .  .  .  .  GT=CNODE(J+1,K+3)
              .  .  .  .  .  .  .  .  GB=CNODE(J,K+3)
              .  .  .  .  .  .  .  .  PMASS=PMASS+DEL*(0.25*(WT*CT*GT+WB*CB*GB)
           1. .  .  .  .  .  .  .  .  +CON*(WB*CT*GT+WT*CB*GT+WT*CT*GB+WT*CB*GB+WB*CT*GB+WB*CB*GT))
           2. .  .  .  .  .  .  .  .  *UPSQ0
              .  .  .  .  .  .  .  .  ...FIN
              .  .  .  .  .  .  ...FIN
              .  .  .  .  .  ...FIN
              .  .  .  .  .  THASS(K)=THASS(K)+CMASS
              .  .  .  .  .  CMASS=CMASS/QHIN(I)
              .  .  .  .  .  CMSAB=CMSAT
              .  .  .  .  .  CMSBB=CMSBT
              .  .  .  .  .  NELMBT=NELMTP
              .  .  .  .  .  IF(K.NE.7)
              .  .  .  .  .  .  PMSAB=PMSAT
              .  .  .  .  .  .  PMSBB=PMSBT
              .  .  .  .  .  .  THASS(K+3)=THASS(K+3)+PMASS
              .  .  .  .  .  .  PMASS=PMASS/QHIN(I)
              .  .  .  .  .  ...FIN
       C      .  .  .  .  .
       C      .  .  .  .  .  NOTE: CMASS IS IN (KG/M**3)
       C      .  .  .  .  .
              .  .  .  .  .  COMPUTE-PROFILE-VALUES
              .  .  .  .  ...FIN
              .  .  .  ...FIN
              .  .  ...FIN
              .  ...FIN
              ...FIN
       C      DISTRIBUTES INITIAL CONDITIONS UPSTREAM OF SUBSEQUENT SEGMENTS BY
       C      CONSERVING MASS FLUX -- ASSUMES LINEAR UPSTREAM DISTRIBUTIONS AND
       C      CONSTANT WIDTHS.
              ELSE
              .  NEWIC=.TRUE.
              .  READ(INFLO)PDEPTH,PDELZ,MELEM,(QHOLD(J),PXSAB(J),PWID(J),J=1,
           1. .  MELEM),((CNODE(J,K),K=1,MAXCON),J=1,MELEM+1),
           2. .  ((OCNODE(J,K),K=1,MAXCON),J=1,MELEM+1)
              .  IF(DEPTH.LE.DEPMIN) GO TO 300
```

```
              .  XS=0.
              .  PXS=0.
              .  DO (I=1,NELEM) XS=XS+XSAREA(I)
              .  DO (I=1,MELEM) PXS=PXS+PXSAR(I)
              .  RATIO=PXS/XS
              .  CALL EQUPXS(PXSAR,PWID,PDELZ,XSAREA,NELEM,MELEM,RATIO,IELP,
             1.  HEQXS)
      C       .  ALLOCATES MASS BY CONSERVING RELATIVE CROSS-SECTIONAL AREAS
              .  DO (K=1,MAXCON)
              .  .  IF(K.LE.3.OR.K.EQ.7)
              .  .  .  K3=K+3
              .  .  .  UPSQO=QHOLD(1)
              .  .  .  CT=(CNODE(2,K)+ OCNODE(2,K))/2.
              .  .  .  CB=(CNODE(1,K) + OCNODE(1,K))/2.
              .  .  .  NELMBT=1
              .  .  .  CMSAB=(CT+CB)/2.*UPSQO
              .  .  .  CMSBB=0.
              .  .  .  TMASS(K)=0.
              .  .  .  IF (K .NE. 7)
              .  .  .  .  GT=(CNODE(2,K3)+ OCNODE(2,K3))/2.
              .  .  .  .  GB=(CNODE(1,K3) + OCNODE(1,K3))/2.
              .  .  .  .  PMSAB=UPSQO*(GT+GB)/2.
              .  .  .  .  PMSBB=0.
              .  .  .  .  TMASS(K3)=0.
              .  .  .  ...FIN
              .  .  .  DO (I=1,NELEM)
              .  .  .  .  NELMTP=IELP(I)
              .  .  .  .  NT=NELMTP+1
              .  .  .  .  NB=NELMTP
              .  .  .  .  ET=NB*PDELZ
              .  .  .  .  EB=(NB-1)*PDELZ
              .  .  .  .  UPSQO=QHOLD(NB)
              .  .  .  .  CT=(CNODE(NT,K) + OCNODE(NT,K))/2.
              .  .  .  .  CB=(CNODE(NB,K) + OCNODE(NB,K))/2.
              .  .  .  .  HEL=HEQXS(I)
              .  .  .  .  FAC1=(HEL-EB)/PDELZ
              .  .  .  .  FAC2=(ET-HEL)/PDELZ
              .  .  .  .  CTOP=CT*FAC1+CB*FAC2
              .  .  .  .  CMSAT=FAC2*(CT+CTOP)*UPSQO/2.
              .  .  .  .  CMSBT=FAC1*(CTOP+CB)/2.*UPSQO
              .  .  .  .  IF(K.NE.7)
              .  .  .  .  .  GT=(CNODE(NT,K3) + OCNODE(NT,K3))/2.
              .  .  .  .  .  GB=(CNODE(NB,K3) + OCNODE(NB,K3))/2.
              .  .  .  .  .  GTOP=GT*FAC1+GB*FAC2
              .  .  .  .  .  PMSAT=FAC2*(GT+GTOP)*UPSQO/2.
              .  .  .  .  .  PMSBT=FAC1*(GTOP+GB)*UPSQO/2.
              .  .  .  .  ...FIN
              .  .  .  .  INDIC=NELMTP-NELMBT
              .  .  .  .  CMASS=0.
              .  .  .  .  PMASS=0.
              .  .  .  .  IF(INDIC.EQ.0)
              .  .  .  .  .  CMASS=CMSBT-CMSBB
              .  .  .  .  .  IF(K.NE.7) PMASS=PMSBT-PMSBB
              .  .  .  .  ...FIN
              .  .  .  .  IF(INDIC.GE.1)
              .  .  .  .  .  CMASS=CMSBT+CMSAB
```

```
          .  .  .  .  .  .  IF(K.NE.7) PMASS=PMSBT+PMSAB
          .  .  .  .  .  ...FIN
          .  .  .  .  .  IF(INDIC.GE.2)
          .  .  .  .  .  .  DO(J=NELMBT+1,NELMTP-1)
          .  .  .  .  .  .  .  CT=(CNODE(J+1,K) + OCNODE(J+1,K))/2.
          .  .  .  .  .  .  .  CB=(CNODE(J,K) + OCNODE(J,K))/2.
          .  .  .  .  .  .  .  Q=QHOLD(J)
          .  .  .  .  .  .  .  CMASS=CMASS+(CT+CB)/2.*Q
          .  .  .  .  .  .  .  IF(K.NE.7)
          .  .  .  .  .  .  .  .  GT=(CNODE(J+1,K3) + OCNODE(J+1,K3))/2.
          .  .  .  .  .  .  .  .  GB=(CNODE(J,K3) + OCNODE(J,K3))/2.
          .  .  .  .  .  .  .  .  PMASS=PMASS+(GT+GB)/2.*Q
          .  .  .  .  .  .  .  ...FIN
          .  .  .  .  .  .  ...FIN
          .  .  .  .  .  ...FIN
          .  .  .  .  .  TMASS(K)=TMASS(K)+CMASS
          .  .  .  .  .  CMASS=CMASS/QHIN(I)
          .  .  .  .  .  CMSAB=CMSAT
          .  .  .  .  .  CMSBB=CMSBT
          .  .  .  .  .  NELMBT=NELMTP
          .  .  .  .  .  IF(K.NE.7)
          .  .  .  .  .  .  TMASS(K3)=TMASS(K3)+PMASS
          .  .  .  .  .  .  PMASS=PMASS/QHIN(I)
          .  .  .  .  .  .  PMSAB=PMSAT
          .  .  .  .  .  .  PMSBB=PMSBT
          .  .  .  .  .  ...FIN
          .  .  .  .  .  COMPUTE-PROFILE-VALUES
          .  .  .  .  ...FIN
          .  .  .  ...FIN
          .  ...FIN
          ...FIN
          RETURN
   200    CONTINUE
          FERROR = .TRUE.
          WRITE(6,1)
   1      FORMAT(10X,'FATAL ERROR - INITIAL CONDITIONS TO SEGMENT 1',
         1 ' HAVE BEEN EXHAUSTED')
          RETURN
   300    CONTINUE
          WRITE(6,2)
          WRITE(6,3) ISEG
   2      FORMAT(10X,'POTENTIAL DIFFICULTY - DEPTH.LE.DEPMIN')
   3      FORMAT(10X,'SEGMENT NUMBER =',I5)
          RETURN

   ----------------------------------------

          TO COMPUTE-PROFILE-VALUES
          .  WHEN(I.EQ.1)
          .  .  WHEN(K.EQ.7)
          .  .  .  CCIN(1,K)=CMASS
          .  .  .  CCIN(2,K)=CMASS
          .  .  ...FIN
          .  .  ELSE
          .  .  .  COEF=0.
          .  .  .  KK=K+3
```

```
00331        .  .  .  WS=VSET(K)*AREA(1)/(AWID(1)*ALEN)
00332        .  .  .  EZ=DFZ(K)
00333        .  .  .  CCIN(1,K)=(2.*CMASS-COEF*DELZ/EZ)/(2.-WS*DELZ/EZ)
00334        .  .  .  CCIN(2,K)=2.*CMASS-CCIN(1,K)
00335        .  .  .  CCIN(1,KK)=(2.*PMASS-COEF*DELZ/EZ)/(2.-WS*DELZ/EZ)
00336        .  .  .  CCIN(2,KK)=2.*PMASS-CCIN(1,KK)
00337        .  .  ...FIN
00338        .  ...FIN
00339        .  ELSE
00340        .  .  CCIN(I+1,K)=2.*CMASS-CCIN(I,K)
00341        .  .  IF (K.NE.7)
00342        .  .  .  CCIN(I+1,K+3)=2.*PMASS-CCIN(I,K+3)
00343        .  .  ...FIN
00344        .  ...FIN
00345        ...FIN
00346        END
```

PROCEDURE CROSS-REFERENCE TABLE

```
00322 COMPUTE-PROFILE-VALUES
      00207  00304
```

(FLECS VERSION 22.46)

```
      ----------------------------------------

      SUBROUTINE INIDAT(ANALMT,ANALYS, DELTH, ECHO, HLDERR, ITPRT,
     1              NSEG, NSTEPS, NUMERR, SIMLEN, DEPMIN)
C
C    THIS ROUTINE READS THE INITIAL DATA COMMON TO ALL SEGMENTS AND IS
C    ONLY CALLED FOR SEGMENT NUMBER 1.
C
C      FORMAL PARAMETERS:
C          ANALMT - ANALYSIS CONCENTRATION LIMIT
C          ANALYS - TIME SERIES ANALYSIS CONTROL VARIABLE (L*1)
C          DELTH  - TIME STEP LENGTH (SECONDS)
C          ECHO   - LINE PRINTER ECHO CONTROL VARIABLE (L*1)
C          HLDERR - HOLDING ARRAY FOR ERROR NUMBERS (BYTE)
C          ITPRT  - PRINT FREQUENCY
C          NSEG   - NUMBER OF SEGMENTS
C          NSTEPS - NUMBER OF TIME STEPS TO BE TAKEN (I*4)
C          NUMERR - NUMBER OF INPUT ERRORS DETECTED
C          SIMLEN - SIMULATION LENGTH (SECONDS - I*4)
C      DEPMIN - MINIMUM (CUTOFF) FLOW DEPTH (METERS)
C
C    CALLED BY: SERATRA
C    CALLES: PUTERR
C
      BYTE HLDERR(100)
C
      INTEGER*4 SIMLEN,NSTEPS
C
      LOGICAL*1 ECHO,ANALYS
C
      DIMENSION TITLE(40)
C
      DATA MAXSEG /35/
      IF (ECHO)
C     .
C     .    *** PRINT HEADING ***
      .    WRITE(6,1)
      .    WRITE(6,2)
      ...FIN
C
C      CARDS 1 AND 2.....SIMULATION IDENTIFICATION TITLE
C
      READ(1,3) (TITLE(I),I=1,40)
      IF (ECHO) WRITE(6,4) (TITLE(I),I=1,40)
C
C      CARD 3............GENERAL INFORMATION COMMON TO ALL SEGMENTS
C
C       COL.  1-10...NSTEPS....NUMBER OF TIME STEPS TO BE TAKEN
C            11-15...NSEG......NUMBER OF SEGMENTS
C            16-20...ITPRT.....PRINT FREQUENCY
C            21-25...ANALYS....TIME SERIES ANALYSIS CONTROL VARIABLE
C            26-35...DELTH.....TIME STEP LENGTH (SECONDS)
C            36-45...ANALMT....LOWER LIMIT OF AVERAGE DISSOLVED
C                              CONCENTRATION, BEFORE THE RESULTS OF A
C                              TIME STEP ARE SAVED, THE AVERAGE
```

```
C                           DISSOLVED CONC. MUST BE > ANALMT.
C          46-55...DEPMIN....MINIMUM (CUTOFF) FLOW DEPTH BELOW WHICH
C                           THE CHANNEL IS CONSIDERED DRIED.
C
           READ(1,5) NSTEPS,NSEG,ITPRT,ANALYS,DELTH,ANALMT,DEPMIN
C
C          *** COMPUTE SIMULATION LENGTH (SECONDS) ***
           SIMLEN = DELTH
           SIMLEN = SIMLEN * NSTEPS
           IF (ECHO)
           .  WRITE(6,6) NSTEPS,NSEG,ITPRT,ANALYS,DELTH,ANALMT,SIMLEN,DEPMIN
           ...FIN
C
           IF (NSTEPS .LE. 0) CALL PUTERR(1,NUMERR,HLDERR)
           IF (NSEG.LE.0 .OR. NSEG.GT.MAXSEG) CALL PUTERR(2,NUMERR,HLDERR)
           IF (ITPRT .LE. 0) CALL PUTERR(3,NUMERR,HLDERR)
           IF (DELTH .LE. 0.0) CALL PUTERR(5,NUMERR,HLDERR)
C
           RETURN
       1   FORMAT(1H0,34X,'SEDIMENT AND CONTAMINANT TRANSPORT SIMULATION',
          1  ' PROGRAM - SERATRA')
       2   FORMAT(1H0,54X,'PROBLEM SPECIFICATIONS')
       3   FORMAT(20A4)
       4   FORMAT(1H0,25X,20A4/26X,20A4)
       5   FORMAT(I10,2I5,L5,2F10.0,E10.3)
       6   FORMAT(1H0,8X,I10,'...NUMBER OF TIME STEPS TO BE TAKEN'/
          1  14X,I5,'...NUMBER OF SEGMENTS'/
          2  14X,I5,'...PRINT FREQUENCY (# OF TIME STEPS)'/
          4  18X,L1,'...TIME SERIES ANALYSIS CONTROL'/
          5  7X,1PE12.5,'...TIME STEP LENGTH (SECONDS)'/
          6  7X,1PE12.5,'...TIME SERIES CONCENTRATION LIMIT'/
          7  9X,I10,'...COMPUTED SIMULATION LENGTH (SECONDS)'/
          8  9X,E10.3,'...MINIMUM (CUTOFF) FLOW DEPTH (METERS)')
C
           END

      (FLECS VERSION 22.46)

      -----------------------------------------
```

```
(FLECS VERSION 22.46)     13-MAR-81   14:10:04   PAGE  0001

        --------------------------------------
C                        [106,1]MSGENT.FLX
C
C       THIS UTILITY PROGRAM IS USED TO ENTER TEXT INTO THE ERROR MESSAGE
C       FILE USED BY SERATRA. EACH ERROR MESSAGE IS TWO RECORDS (160 BYTES)
C       LONG, MADE UP OF 3 PARTS.
C       (1) FATAL OR WARNING TAG (BYTE 1)
C       (2) ERROR IDENTIFICATION NUMBER (BYTES 2-4)
C       (3) MESSAGE TEXT (BYTES 5-160)
C
C       THE DIRECT ACCESS FILE WILL CURRENTLY HOLD 100 SEPARATE ERRORS.
C
        BYTE BUFF(80,2)
C
        OPEN(UNIT=1,NAME='MSGENT.DAT',TYPE='OLD',READONLY)
        OPEN(UNIT=2,NAME='SERATERR.MSG',TYPE='NEW',ACCESS='DIRECT',
       1 MAXREC=200,RECORDSIZE=80,FORM='FORMATTED',
       2 ASSOCIATEVARIABLE=III,INITIALSIZE=25)
C
        NMSG=0
  100   CONTINUE
        READ(1,1,END=200)BUFF(1,1),IND,(BUFF(I,1),I=2,77),
       1 (BUFF(K,2),K=1,80)
        NMSG=NMSG+1
        NREC=(IND-1)*2+1
        WRITE(2'NREC,2)BUFF(1,1),IND,(BUFF(I,1),I=2,77)
        NREC=NREC+1
        WRITE(2'NREC,3)(BUFF(I,2),I=1,80)
        GO TO 100
  200   CONTINUE
        DO (IND=1,NMSG)
        .  NREC=(IND-1)*2+1
        .  READ(2'NREC,3)(BUFF(I,1),I=1,80)
        .  NREC=NREC+1
        .  READ(2'NREC,3)(BUFF(I,2),I=1,80)
        .  WRITE(3,4)BUFF
        ...FIN
        STOP
C
      1 FORMAT(A1,I3,76A1/80A1)
      2 FORMAT(A1,I3,76A1)
      3 FORMAT(80A1)
      4 FORMAT(1X,80A1/1X,80A1)
C
        END

        (FLECS VERSION 22.46)

        --------------------------------------
```

```
      ---------------------------------------
      SUBROUTINE PARTIC(ABAR, ALEN, B, C, CCIN, COLD, DECAY, DELTD,
     1          DFZ, I, J, NBED, NELEM, QHOUT, QHIN, QV, SORBK,
     2           SD, SR, ALFA, BETA, VEL1, VEL2, VSET,
     3 DELZ, DEPTH, BWID, AWID, BETA1, BETA2, DSORB)
C
C    THIS SUBROUTINE CALCULATES COEFFICIENTS OF CONVECTIVE, DECAY AND
C     SOURCE TERMS FOR TRANSPORT OF POLLUTANT ATTACHED TO SEDIMENT.
C
C      INPUT PARAMETERS:
C         ABAR   - AVERAGE ELEMENT AREAS
C         B      - BED CONDITIONS
C         C      - WATER CONDITIONS
C         CCIN   - CONCENTRATION OF INFLOW
C         COLD   - CELL-CENTERED CONCENTRATION
C         DECAY  - FIRST ORDER DECAY
C         DELTD  - TIME STEP (DAYS)
C         DFZ    - DIFFUSION COEFFICIENT
C         DELZ    - ELEMENT THICKNESS
C         I      - ELEMENT INDEX
C         J      - PARAMETER INDEX
C         NBED   - NUMBER OF BED LAYERS
C         NELEM  - NUMBER OF ELEMENTS
C         QHIN   - INFLOW DISCHARGE
C         QV     - VERTICAL DISCHARGE
C         SORBK  - ADSORBTION ON SEDIMENT, (1-3) M**3/KG, (4-9) 1/DAY
C      DSORB  - DESORPTION FROM SEDIMENT, (1-3) M**3/KG, (4-9) 1/DAY
C         SR     - EROSION RATE, KG(PC)/M**3/DAY
C      SD     - DEPOSITION RATE, KG(PC)/M3/DAY
C      OUTPUT PARAMETERS:
C         ALFA   - DECAY TERM, 1/DAY
C         BETA   - SOURCE OR SINK TERM, PC (KG)/M**3/DAY
C      BETA1  - INFLUENT SOURCE TERM FOR I-TH NODE, PC (KG)/M**3/DAY
C         BETA2  - INFLUENT SOURCE TERM FOR I+1 TH NODE, PC (KG)/M**3/DAY
C         VEL1   - FIRST CONVECTIVE TERM, M/DAY
C         VEL2   - SECOND CONVECTIVE TERM, M/DAY
C
C      CALLED BY TRANSP.
C
      INCLUDE 'ELMSIZ.PRM'
C
      DIMENSION ABAR(MXELEM), AREA(MXELEM), B(MAXLEV,MAXCON-1),
     1          CCIN(MXELEM,MAXCON), DECAY(6), DFZ(4), QHIN(MXELEM),
     2          QV(MXELEM), SORBK(9), SR(6), COLD(MXELEM,MAXCON),
     3          C(MXELEM,MAXCON), VSET(3), QHOUT(MXELEM),
     4    SD(MXELEM,6), BWID(MXELEM), AWID(MXELEM), CBAR(MXELEM,MAXCON)
     5    , DSORB(9)
C
C
      ZERO = 1.0E-10
      JM3=J - 3
      IP1 = I + 1
C
C    *** CONVECTIVE TERM ***
```

```
C
             AW = QV(I)
             VEL1 =(AW-VSET(JM3)*BWID(I)*ALEN)/ABAR(I)
C
             AW = QV(IP1)
             VEL2=(AW-VSET(JM3)*BWID(I+1)*ALEN)/ABAR(I)
C
C     *** DECAY TERM ***   (EFFLUENT)
C
             ALFA=QHOUT(I)/(ABAR(I)*DELZ) + DECAY(1)
C
C     *** SOURCE OR SINK TERM ***   (INFLUENT)
C
             BETA1=QHIN(I)/(ABAR(I)*DELZ)*(CCIN(I,J)/3.+CCIN(I+1,J)/6.)
             BETA2=QHIN(I)/(ABAR(I)*DELZ)*(CCIN(I,J)/6.+CCIN(I+1,J)/3.)
C ************************************************************************
C *                                                                      *
C *  WARNING: CCIN IS WRITTEN INTO CBAR AS A FIRST                       *
C *           APPROXIMATION TO THE EVENTUAL AVERAGE                      *
C *           CONCENTRATION.  THE ONLY MEANS BY WHICH                    *
C *           TO ASSURE THE ACCURACY OF THIS                             *
C *           ASSUMPTION IS TO ITERATE TO THE CORRECT                    *
C *           SOLUTION, AND USE NEW ITERATES TO BETTER                   *
C *           APPROXIMATE CBAR.                                          *
C *                                                                      *
C ************************************************************************
C
C     *** ABSORPTION / DESORPTION ***
C
             DO (IE = 1,MXELEM)
             .  DO (IC = 1,MAXCON)
             .  .  CBAR(IE,IC) = CCIN(IE,IC)
             .  ...FIN
             ...FIN
C
C
             IF(CBAR(I,JM3).GT.0.0.AND.CBAR(I+1,JM3).GT.0.0)
             .  ADDS1=      SORBK(J)*SORBK(JM3)/12.*(3.*CBAR(I,JM3)*CBAR(I,7)
            1.  +CBAR(I,JM3)*CBAR(I+1,7)+CBAR(I+1,JM3)*CBAR(I,7)+
            2.  CBAR(I+1,JM3)*CBAR(I+1,7))-SORBK(J)/6.*(2.*CBAR(I,J)+
            3.  CBAR(I+1,J))
             .  DSAD1=      DSORB(J)*DSORB(JM3)/12.*(3.*CBAR(I,JM3)*CBAR(I,7)
            1.  +CBAR(I,JM3)*CBAR(I+1,7)+CBAR(I+1,JM3)*CBAR(I,7)+
            2.  CBAR(I+1,JM3)*CBAR(I+1,7))-DSORB(J)/6.*(2.*CBAR(I,J)+
            3.  CBAR(I+1,J))
             .  ADDS2=      SORBK(J)*SORBK(JM3)/12.*(CBAR(I,JM3)*CBAR(I,7)
            1.  +CBAR(I,JM3)*CBAR(I+1,7)+CBAR(I+1,JM3)*CBAR(I,7)
            2.  +3.*CBAR(I+1,JM3)*CBAR(I+1,7))-SORBK(J)/6.*(CBAR(I,J)
            3.  +2.*CBAR(I+1,J))
             .  DSAD2=      DSORB(J)*DSORB(JM3)/12.*(CBAR(I,JM3)*CBAR(I,7)
            1.  +CBAR(I,JM3)*CBAR(I+1,7)+CBAR(I+1,JM3)*CBAR(I,7)
            2.  +3.*CBAR(I+1,JM3)*CBAR(I+1,7))-DSORB(J)/6.*(CBAR(I,J)
            3.  +2.*CBAR(I+1,J))
             .  IF(ADDS1.GT.0.0.OR.ADDS1.EQ.DSAD1)BETA1=BETA1+ADDS1
             .  IF(DSAD1.LT.0.0)BETA1=BETA1+DSAD1
             .  IF(ADDS2.GT.0.0.OR.ADDS2.EQ.DSAD2)BETA2=BETA2+ADDS2
```

```
(FLECS VERSION 22.46)      28-JAN-82    10:18:31   PAGE  00003

              .  IF(DSAD2.LT.0.0)BETA2=BETA2+DSAD2
              ...FIN
         C
         C
         C    *** SCOUR OR DEPOSITION ***
         C
              BETA = SR(J) - SD(I,J)
              RETURN
              END

         (FLECS VERSION 22.46)

         ----------------------------------------
```

```
      ----------------------------------------
            SUBROUTINE PHOINP(ECHO,JULIAN,KAY1,KAY2,PCOEF)
      C
      C   THIS SUBROUTINE IS RESPONSIBLE FOR READING THE PHOTOLYSIS INPUT
      C   DATA AND COMPUTING THE FIRST TERM OF THE RATE OF CHANGE EQUATION.
      C
      C     FORMAL PARAMETERS:
      C       ECHO   - LINE PRINTER ECHO CONTROL VARIABLE - L*1
      C       JULIAN - JULIAN STARTING DATE OF THE SIMULATION
      C       KAY1   - LIGHT EXTINCTION COEFFICIENT OF WATER
      C       KAY2   - LIGHT EXTINCTION COEFFICIENT OF SUSPENDED SEDIMENTS
      C                IN WATER
      C       PCOEF  - FIRST TERM OF THE RATE OF CHANGE EQUATION
      C
      C   CALLED BY: SERATRA
      C
            INTEGER*4 SECDAY
            LOGICAL*1 ECHO
      C
            REAL KAY1,KAY2
      C
            DIMENSION E(18),SI(18,4),WL(18),PCOEF(4)
      C
            DATA SECDAY/86400/
      C     *** THESE ARE THE SUNLIGHT WAVE LENGTHS THAT DATA IS EXPECTED FOR
            DATA WL / 300.00, 303.75, 308.75, 313.75, 318.75, 323.10,
           1          346.00, 370.00, 400.00, 430.00, 460.00, 490.00,
           2          536.25, 586.50, 637.50, 687.50, 756.00, 800.00/
      C
      C     ....FIRST DATA SET
      C          COL.  1- 5...JULIAN....JULIAN STARTING DATE OF THE SIMULATION
      C                                 WHEN THIS IS INPUT AS A ZERO, NO
      C                                 PHOTOLYSIS CALCULATIONS ARE MADE.
      C                6-15...PHI.......THE REACTION QUANTUM YIELD FOR THE
      C                                 CHEMICAL IN AIR-SATURATED, PURE WATER.
      C                                 A MEASURE OF THE EFFICIENCY WITH WHICH
      C                                 A PHOTOCHEMICAL PROCESS CONVERTS
      C                                 ADSORBED LIGHT INTO CHEMICAL REACTION.
      C               16-25...KAY1......LIGHT EXTINCTION COEFFICIENT OF WATER
      C                                 (1/M)
      C               26-35...KAY2......LIGHT EXTINCTION COEFFICIENT OF
      C                                 SUSPENDED SEDIMENTS IN WATER (SELF-
      C                                 SHADING COEFFICIENT) M**4/KG**2
      C
            READ(1,1) JULIAN,PHI,KAY1,KAY2
            WHEN (JULIAN .EQ. 0)
            .  DO (I=1,4) PCOEF(I) = 0.0
            .  PHI = 0.0
            .  KAY1 = 0.0
            .  KAY2 = 0.0
            .  IF (ECHO) WRITE(6,2)
            ...FIN
            ELSE
      C     .
```

```
C       .  ....SECOND DATA SET - ADSORBTION COEFFICIENTS TABLE
C       .        THIS IS THE TABLE OF ADSORTION COEFFICIENTS FOR 18
C       .        DIFFERENT WAVELENGTHS THAT ARE A MEASURE OF THE CHEMICAL'S
C       .        ABILITY TO ADSORB LIGHT AT THE DIFFERENT WAVELENGTHS.
C       .        WAVELENGTH UNITS ARE NANO METERS.
C       .    ....RECORD 1
C       .         COL.  1-10...E( 1)...COEF. FOR WAVELENGTH OF 300.00
C       .              11-20...E( 2)...COEF. FOR WAVELENGTH OF 303.75
C       .              21-30...E( 3)...COEF. FOR WAVELENGTH OF 308.75
C       .              31-40...E( 4)...COEF. FOR WAVELENGTH OF 313.75
C       .              41-50...E( 5)...COEF. FOR WAVELENGTH OF 318.75
C       .              51-60...E( 6)...COEF. FOR WAVELENGTH OF 323.10
C       .              61-70...E( 7)...COEF. FOR WAVELENGTH OF 346.00
C       .              71-80...E( 8)...COEF. FOR WAVELENGTH OF 370.00
C       .
C       .   ....RECORD 2
C       .        COL.  1-10...E( 9)...COEF. FOR WAVELENGTH OF 400.00
C       .             11-20...E(10)...COEF. FOR WAVELENGTH OF 430.00
C       .             21-30...E(11)...COEF. FOR WAVELENGTH OF 460.00
C       .             31-40...E(12)...COEF. FOR WAVELENGTH OF 490.00
C       .             41-50...E(13)...COEF. FOR WAVELENGTH OF 536.25
C       .             51-60...E(14)...COEF. FOR WAVELENGTH OF 587.50
C       .             61-70...E(15)...COEF. FOR WAVELENGTH OF 637.50
C       .             71-80...E(16)...COEF. FOR WAVELENGTH OF 687.50
C       .
C       .   ....RECORD 3
C       .        COL.  1-10...E(17)...COEF. FOR WAVELENGTH OF 756.00
C       .             11-20...E(18)...COEF. FOR WAVELENGTH OF 800.00
C       .
        .  READ(1,4) (E(L),L=1,18)
C       .
C       .  ....THIRD DATA SET - SOLAR INTENSITY TABLE
C       .        THIS TABLE CONSISTS OF FOUR SETS OF 18 VALUES. THE
C       .        FOUR SETS CORRESPOND TO SPRING, SUMMER, FALL, AND WINTER,
C       .        RESPECTIVELY. THE 18 VALUES CORRESPOND TO THE 18 WAVELENGTHS
C       .        AS DESCRIBED ABOVE IN THE ADSORPTION COEFFICIENT TABLE.
C       .        THE INCLUSIVE DATES FOR EACH SEASON ARE GIVEN BELOW:
C       .                   CALENDER DATES          JULIAN DATES
C       .                   --------------          ------------
C       .       SPRING    MARCH 1 - MAY 31            60-151
C       .       SUMMER    JUNE 1  - AUG. 31          152-243
C       .       FALL      SEP. 1  - NOV. 30          244-334
C       .       WINTER    DEC. 1  - FEB. 28          335-365; 1-59
C       .
        .  READ(1,4) ((SI(L,I),L=1,18),I=1,4)
C       .
        .  IF (ECHO)
        .  .  WRITE(6,3)
        .  .  WRITE(6,7) JULIAN,PHI,KAY1,KAY2
        .  .  WRITE(6,5)
        .  .  DO (L=1,18)
        .  .  .  WRITE(6,6) WL(L),E(L),(SI(L,I),I=1,4)
        .  .  ...FIN
        .  ...FIN
C       .
C       .  *** THE JULIAN DATE IS ADJUSTED TO MAKE THE FIRST DAY OF
```

```
C       .        SPRING JULIAN DAY 1 ***
        .   WHEN (JULIAN .GE. 60) JULIAN = JULIAN - 59
        .   ELSE JULIAN = JULIAN + 306
C       .
C       .   *** COMPUTE THE FIRST TERM OF THE RATE OF CHANGE EQUATION
C       .        FOR EACH OF THE FOUR SEASONS ***
        .   DO (I=1,4)
        .   .  PCOEF(I) = 0.0
        .   .  DO (L=1,18) PCOEF(I) = PCOEF(I) + E(L) * SI(L,I)
        .   .  PCOEF(I) = PHI * PCOEF(I) * (2.303/6.02E20) * SECDAY
        .   ...FIN
        ...FIN
        RETURN
C
1       FORMAT(I5,3F10.0)
2       FORMAT(1H0,13X,'NO PHOTOLYSIS DEGRADATION WILL BE COMPUTED')
3       FORMAT(1H0,48X,'PHOTOLYSIS TABLES AND COEFFICIENTS')
4       FORMAT(8F10.0/8F10.0/2F10.0)
5       FORMAT(1H0,56X,'PHOTOLYSIS TABLES'/39X,'ADSORBTION',
       1 23X,'SOLAR INTENSITIES'/23X,'LAMDA CENTER',
       2 2X,'COEFFICIENTS',5X,'SPRING',8X,'SUMMER',9X,'FALL',
       3 9X,'WINTER'/23X,6(12('-'),2X))
6       FORMAT(26X,F6.2,3X,5(2X,1PE12.4))
7       FORMAT(26X,I5,'...JULIAN STARTING DATE'/
       1 14X,1PE12.5,'...REACTION QUANTUM YIELD'/
       2 14X,1PE12.5,'...LIGHT EXTINCTION COEFFICIENT OF WATER'/
       3 14X,1PE12.5,'...LIGHT EXTINCTION COEFFICIENT OF SUSPENDED',
       4                ' SOLIDS IN WATER')
C
        END
```

(FLECS VERSION 22.46)

```
      ------------------------------------
      SUBROUTINE PROFIL(ALEN, AWID, DELZ, DEPTH, NELEM, Q,
     1                  USTAR, VOL, QH, EL, IFLAG, DELTA)
C
C   THIS ROUTINE ASSIGNS A LOGARITHMIC OR UNIFORM PROFILE FOR THE
C   BULK VOLUMETRIC FLOWS.
C
C       INPUT PARAMETERS:
C          ALEN   - SEGMENT LENGTH
C          AWID   - SEGMENT WIDTH
C          DELZ   - STANDARD ELEMENT THICKNESS
C          DEPTH  - FLOW DEPTH
C          NELEM  - NUMBER OF VERTICAL ELEMENTS IN THE SEGMENT
C          Q      - FLOW TO BE DISTRIBUTED
C          USTAR  - SHEAR VELOCITY
C          VOL    - SEGMENT VOLUME
C       OUTPUT PARAMETERS:
C          QH     - DISTRIBUTED FLOW
C
C       CALLED BY: HYDFLO, ICFLO
C
      INCLUDE 'ELMSIZ.PRM'
C
      DIMENSION QH(MXELEM), Z(MXELEM), Z1(MXELEM), AWID(MXELEM),
     1          EL(MXELEM)
      DATA XK/0.4/
      DATA G/9.81/
C
      DO (I = 1,MXELEM)
      .  QH(I) = 0.0
      ...FIN
      UBAR=Q / VOL * ALEN
C     ZO=DEPTH/(10.**(UBAR*XK/(2.3*USTAR) + 1./2.3))
C
      ICOUNT=0
      ZO=0.001
      C1=UBAR*XK/USTAR+1.0
      CON=DEPTH*(C1-2.303*ALOG10(DEPTH))
      REPEAT UNTIL(ABS(EPS).LT.0.01.OR.ICOUNT.GT.10)
      .  ICOUNT=ICOUNT+1
      .  FZO=CON-ZO*C1+2.303*DEPTH*ALOG10(ZO)
      .  FPZO=DEPTH/ZO-C1
      .  ZP=ZO-FZO/FPZO
      .  EPS=(ZP-ZO)/ZP
      .  ZO=ZP
      .  IF(ZO.LT.0.0)
      .  .  ZO=DEPTH/(10.**(UBAR*XK/(2.303*USTAR)+1./2.303))
      .  .  ICOUNT=11
      .  ...FIN
      ...FIN
C
C
      WHEN (ZO .GT. DELZ/4.0.OR.NELEM.EQ.1)
C     .
```

```
C      .  *** DISTRIBUTE VELOCITY UNIFORMLY ***
       .  WHEN(IFLAG.EQ.0)
       .  .  DO (I=1,NELEM)
       .  .  .  QH(I)=UBAR*DELZ*AWID(I)
       .  .  ...FIN
       .  ...FIN
       .  ELSE
       .  .  DO (I=1,NELEM-1)
       .  .  .  QH(I)=UBAR*(EL(I+1)-EL(I))*AWID(I)
       .  .  ...FIN
       .  .  QH(NELEM)=UBAR*DELTA*AWID(NELEM)
       .  ...FIN
       ...FIN
       ELSE
C      .  *** DISTRIBUTE VELOCITY INTO A LOGARITHMIC PROFILE ***
       .  NUMB=NELEM+1
       .  Z(1)=Z0
       .  Z(NUMB)=DEPTH
       .  Z1(NUMB)=0.
C      .
       .  JX = 1
       .  J=NUMB
       .  REPEAT UNTIL (JX .EQ. NELEM .OR. Z(J) .LE. Z0)
       .  .  JX = JX + 1
       .  .  J=NELEM - JX + 2
       .  .  WHEN (IFLAG.EQ.0) Z1(J)=Z1(J+1)+DELZ
       .  .  ELSE
       .  .  .  WHEN (J.EQ.NELEM) Z1(J)=Z1(J+1)+DELTA
       .  .  .  ELSE Z1(J)=Z1(J+1)+EL(J+1)-EL(J)
       .  .  ...FIN
       .  .  Z(J)=DEPTH - Z1(J)
       .  ...FIN
       .  WHEN (Z(J) .GT. Z0) J = 1
       .  ELSE
       .  .  J1 = J-1
       .  .  DO (I=1,J1)
       .  .  .  QH(I)=0.
       .  .  ...FIN
       .  ...FIN
C      .
C      .
       .  A=2.303 * USTAR / XK
       .  SUM=0.
C      .  XLOGD=ALOG10(DEPTH)
C      .  DO (I=J,NELEM)
C      .  IP1=I+1
C      .  DLZ = Z(IP1) - Z(I)
C      .  T1=A*DLZ * XLOGD
C      .  T2 =A*(Z(IP1) * ALOG10(Z(IP1)) - Z(I) * ALOG10(Z(I))-DLZ)
C      .  QH(I)=(UBAR*DLZ-T1+T2)*AWID(I)
C      .  SUM = SUM + QH(I)
C      .  FIN
C      .
       .  T1=A/(DEPTH-Z0)*(Z0*ALOG10(Z0)-DEPTH*ALOG10(DEPTH))+UBAR
       .  DO (I=J,NELEM)
       .  .  IP1=I+1
```

```
(FLECS VERSION 22.46)     28-JAN-82   10:17:41   PAGE  00003

        .  .  T2=A*(Z(IP1)*ALOG10(Z(IP1))-Z(I)*ALOG10(Z(I)))
        .  .  QH(I)=AWID(I)*(T2+(Z(IP1)-Z(I))*T1)
        .  .  SUM=SUM+QH(I)
        .  ...FIN
C       .
        .   DO (I=1,NELEM)
        .  .      QH(I)=QH(I)/SUM * Q
        .  ...    FIN
        ...FIN
        RETURN
        END

   (FLECS VERSION 22.46)

   ----------------------------------------
```

```
----------------------------------------

      SUBROUTINE PUTERR(ID, NUMERR, HLDERR)
C
C     WHEN AN ERROR IS DETECTED IN THE INPUT STREAM ,
C     THIS SUBROUTINE IS CALLED TO PLACE THE ERROR IDENTIFICATION
C     CODE (ID) INTO THE HOLDING ARRAY (HLDERR) AND INCREMENTS
C     THE NUMBER OF ERRORS (NUMERR).
C
C       CALLED BY: DIMDAT, INIDAT, TRBDAT, UPSDAT
C
      BYTE HLDERR(100)
C
      NUMERR=NUMERR+1
      HLDERR(NUMERR)=ID
      RETURN
      END

(FLECS VERSION 22,46)

----------------------------------------
```

```
      SUBROUTINE RADIUS ( ALEN, AREA, CROSEC, DEPTH, EL, HRAD)
C
C     THIS ROUTINE CALCULATES THE HYDRAULIC RADIUS OF A CROSS-SECTION.
C
C     INPUT PARAMETERS:
C          ALEN   - SEGMENT LENGTH
C          AREA   - SURFACE AREA AT NODAL DEPTHS
C          DEPTH  - DEPTH OF CROSS-SECTION
C          EL     - ELEVATION OF NODAL AREAS
C     OUTPUT PARAMETERS:
C          CROSEC - TOTAL CROSS-SECTIONAL AREA
C          HRAD   - HYDRAULIC RADIUS
C
C     CALLED BY: HYDDAT, ICFLO
C
      INCLUDE 'ELMSIZ.PRM'
      DIMENSION AREA(MXELEM), EL(MXELEM)
C
      CROSEC = 0.
      BLEN = 1./ALEN
      WETPER = AREA(1)*BLEN
      ELBTM = EL(1)
      WBTM = AREA(1)*BLEN
C
      DO (I=2,MXELEM)
      .  ELTOP = EL(I)
      .  WTOP = AREA(I)*BLEN
      .  IF (ELTOP.GE.DEPTH) GO TO 10
      .  CROSEC = (ELTOP - ELBTM)*(WTOP + WBTM)/2. + CROSEC
      .  WETPER = SQRT((ELTOP - ELBTM)**2 + ((WTOP - WBTM)/2.)**2) * 2.0
     1.         + WETPER
      .  ELBTM = ELTOP
      .  WBTM = WTOP
      ...FIN
      WRITE (6,1) CROSEC, WETPER, I
    1 FORMAT ('1', '   ERROR IN SUBROUTINE RADIUS'/
     1             '0  CROSEC = ', E12.4,'M**3'/
     2             '0  WETTED PERIMETER = ', E12.4,'M**2')
      GO TO 20
   10 WTOP = WBTM + (WTOP - WBTM) * (DEPTH - ELBTM)/(ELTOP - ELBTM)
      ELTOP = DEPTH
      CROSEC = CROSEC + (ELTOP - ELBTM) * (WTOP + WBTM)/2.
      WETPER = SQRT ((ELTOP - ELBTM)**2 + ((WTOP -WBTM)/2.)**2) * 2.0
     1        + WETPER
   20 HRAD = CROSEC / WETPER
      RETURN
      END
```

(FLECS VERSION 22.46)

```
      ------------------------------------------

      SUBROUTINE ROSFLO(ALEN, AREA, AWID,C, DELZ, DFZ, NELEM, NELEMP,
     1 PDELZ, PWID, PXSAR, VSET, XSAREA)
C
C   THIS SUBROUTINE IS CALLED EACH TIME STEP (EXCEPT THE FIRST) WHEN THE
C   DEPTH WITHIN THE SEGMENT HAS CHANGED. ITS TASK IS TO REDISTRIBUTE
C   THE CONCENTRATIONS.
C
C   FORMAL PARAMETERS:
C     AWID   - WIDTH @ CURRENT TIME STEP
C     C      - THE NELMPT+1 NODAL CONCENTRATIONS THAT ARE TO BE
C              REDISTRIBUTED
C     DELZ   - THE STANDARD ELEMENT THICKNESS FOR THE CURRENT TIME STEP
C     DFZ    - DIFFUSION-DISPERSION COEFFICIENT
C     NELEM  - THE NUMBER OF ELEMENTS FOR THE CURRENT TIME STEP
C     NELEMP - THE NUMBER OF ELEMENTS DURING THE PREVIOUS TIME STEP
C     PDELZ  - THE STANDARD ELEMENT THICKNESS USED DURING THE PREVIOUS
C              TIME STEP
C     PXSAR  - PRESS SECTION @ PREVIOUS TIME STEP
C     PWID   - WIDTH @ PREVIOUS TIME STEP
C     XSAREA - CROSS SECTION @ CURRENT TIME STEP
C     VSET   - SETTLING VELOCITY OF SEDIMENT
C
C   CALLED BY: SERATRA
C
      INCLUDE 'ELMSIZ.PRM'
C
      DIMENSION AREA(MXELEM), AWID(MXELEM), C(MXELEM,MAXCON),
     1CP(MXELEM,MAXCON), DFZ(4), PWID(MXELEM), PXSAR(MXELEM),
     2VSET(3), XSAREA(MXELEM), IELP(MXELEM), HEQXS(MXELEM), TMASS(7)
C
      PXS = 0.
      XS  = 0.
C
      DO (I=1,NELEM) XS = XS + XSAREA(I)
      DO (I=1,NELEMP) PXS = PXS + PXSAR(I)
      DO (I=1,NELEMP+1)
      .  DO (J=1,MAXCON)
      .  .  CP(I,J) = C(I,J)
      .  ...FIN
      ...FIN
      RATIO = PXS/XS
      CALL EQUPXS(PXSAR, PWID, PDELZ, XSAREA, NELEM, NELEMP, RATIO,
     1IELP, HEQXS)
      DO (K=1,MAXCON)
      .  TMASS(K)=0.
      .  CT=CP(2,K)
      .  CB=CP(1,K)
      .  NELMBT=1
      .  CMASAB=(CT+CB)/2.*PXSAR(1)
      .  CMASBB=0.
      .  DO(I=1,NELEM)
      .  .  NELMTP=IELP(I)
      .  .  NB=NELMTP
```

```
          .  .  NI=NB+1
          .  .  ET=NB*PDELZ
          .  .  EB=ET-PDELZ
          .  .  CT=CP(NT,K)
          .  .  CB=CP(NB,K)
          .  .  HEL=HEQXS(I)
          .  .  FAC1=(HEL-EB)/PDELZ
          .  .  FAC2=(ET-HEL)/PDELZ
          .  .  CTOP=CT*FAC1+CB*FAC2
          .  .  CMASAT=(CT+CTOP)/2.*PWID(NB)*(ET-HEL)
          .  .  CMASBT=(CTOP+CB)/2.*PWID(NB)*(HEL-EB)
          .  .  INDIC=NELMTP-NELMBT
          .  .  CMASS=0.
          .  .  IF(INDIC.EQ.0) CMASS=CMASBT-CMASBB
          .  .  IF(INDIC.GT.0) CMASS=CMASBT+CMASAB
          .  .  IF(INDIC.GT.1)
          .  .  .  DO(J=NELMBT+1,NELMTP-1)
          .  .  .  .  CT=CP(J+1,K)
          .  .  .  .  CB=CP(J,K)
          .  .  .  .  CMASS=CMASS+PXSAR(J)*(CT+CB)/2.
          .  .  .  ...FIN
          .  .  ...FIN
          .  .  TMASS(K)=TMASS(K)+CMASS
    C     .  .  NODAL VALUE IN BOTTOM ELEMENT
          .  .  WHEN(I.EQ.1)
          .  .  .  WHEN(K.NE.7)
    C     .  .  .  .
    C     .  .  .  .  SAND SILT CLAY
    C     .  .  .  .
          .  .  .  .  WHEN(K.LT.4)
          .  .  .  .  .  COEF =0.
          .  .  .  .  .  WS=VSET(K)*AREA(1)/(AWID(1)*ALEN)
          .  .  .  .  .  EZ=DFZ(K)
          .  .  .  .  ...FIN
    C     .  .  .  .
    C     .  .  .  .  POLLUTANT ASSOCIATED WITH SAND SILT CLAY
    C     .  .  .  .
          .  .  .  .  ELSE
          .  .  .  .  .  COEF =0.
          .  .  .  .  .  WS=VSET(K-3)*AREA(1)/(AWID(1)*ALEN)
          .  .  .  .  .  EZ=DFZ(K-3)
          .  .  .  .  ...FIN
    C     .  .  .  .
    C     .  .  .  .  NOTE: CMASS IS IN (KG/M)
    C     .  .  .  .
          .  .  .  .  C(1,K)=(2.*CMASS/XSAREA(1)-COEF*DELZ/EZ)/(2.-WS*DELZ/EZ)
          .  .  .  .  C(2,K)=2.*CMASS/XSAREA(1)-C(1,K)
          .  .  .  ...FIN
    C     .  .  .
    C     .  .  .  DISSOLVED POLLUTANT
    C     .  .  .
          .  .  .  ELSE
          .  .  .  .  C(1,K)=CMASS/XSAREA(I)
          .  .  .  .  C(2,K)=C(1,K)
          .  .  .  ...FIN
          .  .  ...FIN
```

```
C      .  .  NODAL VALUES ABOVE BOTTOM ELEMENT
       .  .  ELSE C(I+1,K) = 2.*CMASS/XSAREA(I)-C(I,K)
       .  .  CMASAB=CMASAT
       .  .  CMASBB=CMASBT
       .  .  NELMBT=NELMTP
       .  ...FIN
       ...FIN
       RETURN
       END
```

(FLECS VERSION 22.46)

```
(FLECS VERSION 22.46)      13-MAR-81   13:28:34  PAGE  00001

         --------------------------------------

               SUBROUTINE RPTERR(NUMERR, HLDERR, FERROR)
         C
         C   THIS ROUTINE IS RESPONSIBLE FOR REPORTING ANY INPUT ERRORS
         C   UNCOVERED BY THE INPUT ROUTINES AND DETERMINING THEIR SEVERITY,
         C
         C     CALLED BY: SERATRA, STRTUP
         C
               BYTE HLDERR(100),BUFF(80,2)
         C
               LOGICAL*1 FERROR
         C
         C     CALLED BY: SERATRA
         C
               OPEN(UNIT=10,NAME='SERATERR.MSG',TYPE='OLD',ACCESS='DIRECT',
              1 FORM='FORMATTED',MAXREC=200,RECORDSIZE=80,
              2 ASSOCIATEVARIABLE=III,READONLY)
         C
               NUMW=0
               NUMF=0
               WRITE(6,4)
               DO (I=1,NUMERR)
               .  IND=(HLDERR(I)-1)*2+1
               .  READ(10'IND,1)(BUFF(K,1),K=1,80)
               .  IND=IND+1
               .  READ(10'IND,1)(BUFF(K,2),K=1,80)
               .  IF(BUFF(1,1).EQ.'W')NUMW=NUMW+1
               .  IF(BUFF(1,1).EQ.'F')NUMF=NUMF+1
               .  WRITE(6,2)((BUFF(K,J),K=1,80),J=1,2)
               ...FIN
               WRITE(6,3)NUMW,NUMF
               CLOSE(UNIT=10)
               IF (NUMF .GT. 0) FERROR = .TRUE.
               RETURN
         C
            1  FORMAT(80A1)
            2  FORMAT(1H0,A1,2X,3A1,3X,76A1/10X,80A1)
            3  FORMAT(//,10X,I5,' WARNING DIAGNOSTICS'/11X,I5,' FATAL ERROR(S)')
            4  FORMAT('/***** DIAGNOSTIC SUMMARY *****')
         C
               END

         (FLECS VERSION 22.46)

         --------------------------------------
```

```
        ----------------------------------------

              SUBROUTINE SAND(ABAR, ALEN, AREA, B, BDIV,  CCIN, DELTD, DELZ,
             1                 DENS, D50, HRAD, NBED, NELEM, POR, QHIN,
             2                 QHOUT, SCSHR, SLOPE, SMETH, STRESS, TEMPR,
             3                 VSET, VOL, XYSO, DEPO, ILAYR, SD, SR,
             4                 XNT, COLD, C, CROSEC, BWID, ECHO7, SCOUR)
        C
        C     THIS SUBROUTINE COMPUTES THE SOURCE/SINK TERMS REQUIRED FOR
        C     SCOUR/DEPOSITION OF SAND.  TRANSPORT CAPACITY IS CALCULATED
        C     ONCE PER SEGMENT.
        C
        C     INPUT PARAMETERS:
        C          ABAR   - AVERAGE VERTICAL PROJECTION AREA
        C          ALEN   - SEGMENT LENGTH
        C          AREA   - ELEMENT (REAL) VERTICAL PROJECTION AREA @ DATA NODES
        C          B      - BED CONDITIONS
        C          BDIV   - STANDARD BED LAYER THICKNESS
        C          BWID   - REAL WIDTH AT CROSS-SECTION BREAK POINTS
        C          C      - WATER CONDITIONS
        C          CCIN   - CONCENTRATION OF INFLOW
        C          CROSEC - TOTAL CROSS-SECTIONAL AREA, M**2
        C          DELTD  - TIME STEP (DAYS)
        C          DELZ   - STANDARD ELEMENT THICKNESS
        C          DELZT  - THICKNESS OF THE TOP ELEMENT
        C          DENS   - DENSITY
        C          D50    - MEDIAN BED SEDIMENT DIAMETER (METER)
        C          HRAD   - HYDRAULIC RADIUS
        C          NBED   - NUMBER OF BED LAYERS
        C          NELEM  - NUMBER OF ELEMENTS
        C          POR    - POROSITY
        C          QHIN   - INFLOW DISCHARGE
        C          QHOUT  - OUTFLOW DISCHARGE
        C          SCSHR  - CRITICAL SHEAR STRESS FOR SCOUR
        C          SLOPE  - ENERGY OR RIVER BED SLOPE
        C          SMETH  - METHOD TO BE USED WHEN COMPUTING SAND CAPACITY (BYTE
        C                   =T: TOFFALETTI'S METHOD
        C                   =C: COLBY'S METHOD
        C          STRESS - BED SHEAR STRESS
        C          TEMPR  - WATER TEMPERATURE
        C          VSET   - PARTICLE SETTLING VELOCITY
        C          VOL    - VOLUME
        C          XYSO   - THICKNESS OF TOP BED LAYER
        C       OUTPUT PARAMETERS:
        C          ILAYR  - NO. OF BED LAYERS AFFECTED BY SED. DEPOSITION AND ER
        C          SD     - DEPOSITION RATE, (KG(PC)/M**3/DAY)
        C          SR     - EROSION RATE, (KG(PC)/M**3/DAY)
        C          XNT    - WEIGHT OF TOP BED SEDIMENT LAYER, (KG/M**2)
        C       DEPO   - BED DEPOSITION RATE (KG(PC)/M2/DAY)
        C       SCOUR  - BED SCOUR RATE (KG(PC)/M2/DAY)
        C
        C       CALLED BY: TRANSP.
        C       CALLS: TOFFAL, COLBY
        C
              INCLUDE 'ELMSIZ.PRM'
```

```
C
          BYTE SMETH
C
          LOGICAL*1 FERROR, ECHO7
C
C
          DIMENSION ABAR(MXELEM), AREA(MXELEM), B(MAXLEV,MAXCON-1),
         1          COLD(MXELEM,MAXCON),CCIN(MXELEM,MAXCON), DENS(3),
         2          QHIN(MXELEM), QHOUT(MXELEM), SCSHR(3),
         3          SD(MXELEM,6), SR(6), VSET(3), XNT(3), EE(MXELEM),
         4          ILAYR(3), C(MXELEM,MAXCON), BWID(MXELEM),
         5          DEPO(6), SCOUR(6)
C
C
C         INITIALIZE SCALAR AND ARRAY VARIABLES
          ILAYR(1) = 0
          RS=0.0
          CS=0.0
          DEPO(1) =0.0
          DEPO(4) =0.0
          SRC = 0.0
          GSI = 0.0
          QTOT = 0.0
          SR(1)=0.0
          SR(4)=0.0
          SCOUR(1)= 0.0
          SCOUR(4)= 0.0
          RATE = 0.0
          VPCROS = ALEN * BWID(1)
          VOLUME = CROSEC * ALEN
          DO (K=1,NELEM)
          .  SD(K,1)=0.0
          .  SD(K,4)=0.0
          ...FIN
          XNT(1) = 0.
          IF(NBED.NE.0)
          .  XNT(1)=(1.0-POR)/(B(NBED,1)/DENS(1)+B(NBED,2)/DENS(2)+
         1.       B(NBED,3)/DENS(3))*XYSO*B(NBED,1)
          ...FIN
          IF (ECHO7)
C         .
C         .  CALCULATE ACTUAL SAND TRANSPORT WITHIN THE RIVER REACH, SRC(KG/DAY)
C         .
C         .       NOTE: CCIN IS A TIME AVERAGED QUANTITY, COLD IS NOT.   AN
C         .             ITERATIVE LOOP IS CALLED FOR WHERE SRC IS UPDATED UNTIL
C         .             RESULTS ARE UNCHANGED.
C         .
          .  DO (IX=1,NELEM)
          .  .  SRC=SRC+QHIN(IX)*(CCIN(IX,1)+CCIN(IX+1,1))/4.
         1.  .        +QHOUT(IX)*COLD(IX,1)/2.
          .  .  QTOT=QTOT+(QHIN(IX)+QHOUT(IX))/2.
          .  ...FIN
C         .
C         .  IT IS IMPLICITLY ASSUMED THAT A DOWNSTREAM COURANT NUMBER
C         .  AT OR NEAR UNITY IS EMPLOYED IN THIS ANALYSIS
C         .
```

```
C       .
C       .  CALCULATIONS OF STREAM CAPACITY FOR SAND
        .  SELECT (SMETH)
        .  .  ('T')
C       .  .  .
C       .  .  .  THE TOFFALETI TECHNIQUE MAY NOT BE WELL SUITED FOR CHANNEL
C       .  .  .  CROSS-SECTIONS WHICH DIFFER MARKEDLY FROM RECTANGULAR.
C       .  .  .
        .  .  .  CALL TOFFAL(ALEN, D50, GSI, HRAD, QTOT, SLOPE, TEMPR, VOL, VSET,
       1.  .  .              GSU, GSM, GSL, GSB, YU, YM, YL)
        .  .  ...FIN
        .  .  ('C')
        .  .  .  CALL COLBY(ALEN, C, DELZ, D50, HRAD, NELEM, QTOT, TEMPR,
       1.  .  .              VOL, GSI, FERROR)
        .  .  .  IF(FERROR)
        .  .  .  .  CALL TOFFAL(ALEN, D50, GSI, HRAD, QTOT, SLOPE, TEMPR, VOL, VSET,
       1.  .  .  .              GSU, GSM, GSL, GSB, YU, YM, YL)
        .  .  .  .  FERROR = .FALSE.
        .  .  .  ...FIN
        .  .  ...FIN
        .  ...FIN
        .  GSI = GSI * BWID(NELEM + 1)
C       .
C       .  DETERMINE IF DEPOSITION, SCOUR, OR NEITHER OCCURS.
C       .
        .  DIF = GSI - SPC
        .  IF (DIF) 50, 100, 150
C       .
C       .  SAND WITHIN THE WATER COLUMN EXCEEDS CAPACITY - DEPOSITION OCCURS
C       .
50      .  DEPO(1) = -DIF/ AREA(1)
        .  RATE = -DIF/VOLUME
        .  ILAYR(1) = -1
        .  DO (K=1,NELEM)
        .  .  SD(K,1) = RATE
        .  .  VOLK = ABAR(K)*DELZ
        .  .  SED=QHIN(K)*(CCIN(K,1)+CCIN(K+1,1))/4.+QHOUT(K)*COLD(K,1)/2.
        .  .  CONT=QHIN(K)*(CCIN(K,4)+CCIN(K+1,4))/4.+QHOUT(K)*COLD(K,4)/2.
        .  .  RATEK = RATE * VOLK
        .  .  SD(K,4) = RATEK * CONT/SED/VOLK
        .  .  DEPO(4) = DEPO(4)+SD(K,4)*VOLK/AREA(1)
        .  ...FIN
        .  RETURN
C       .
C       .  CAPACITY EQUALS LOAD - NOTHING EXCHANGES
C       .
100     .  RETURN
C       .
C       .  CAPACITY IS GREATER THAN LOAD - SCOUR OCCURS
C       .
C       .
C       .  DETERMINE WHICH (IF ANY)  BED LAYERS ARE SCOURED
C       .
150     .  TRSUSP = DIF * DELTO
        .  ILAYR(1) = 0
        .  IF (NBED.EQ.0) GO TO 200
```

```
         .  DO (K=1,NBED)
         .  .  NB = NBED - K + 1
         .  .  TDENS = (1.0 - POR) / (B(NB,1)/DENS(1) + B(NB,2)/DENS(2)
        1.  .                          + B(NB,3)/DENS(3))
         .  .  DEL = BDIV
         .  .  IF (NB.EQ.NBED) DEL = XYSO
         .  .  TINBED = TDENS * DEL * B(NB,1) * VPCROS
         .  .  WHEN (TRSUSP.GE.TINBED)
         .  .  .  RS = RS + TINBED
         .  .  .  CS = CS + TINBED * B(NB,4)
         .  .  .  TRSUSP = TRSUSP - TINBED
         .  .  .  TINBED = 0.0
         .  .  .  ILAYR(1) = ILAYR(1) + 1
         .  .  .  IF(ILAYR(1).EQ.NBED) GO TO 175
         .  .  ...FIN
         .  .  ELSE
         .  .  .  RS = RS + TRSUSP
         .  .  .  CS = CS + TRSUSP * B(NB,4)
         .  .  .  TINBED = TINBED - TRSUSP
         .  .  .  GO TO 175
         .  .  ...FIN
         .  ...FIN
  175    .  SCOUR(1) = RS / DELTD/ AREA(1)
         .  SCOUR(4) = CS / DELTD/ AREA(1)
         .  XNT(1) = TINBED / VPCROS
         .  SR(1) = RS  /DELTD / VOLUME
         .  SR(4) = CS / DELTD / VOLUME
         ...FIN
  200    RETURN
         END

  (FLECS VERSION 22.46)

  -----------------------------------------
```

```
        --------------------------------------
          SUBROUTINE SAVEIT(B, BDIV, BED, ELEV, C, DELZ,  NBED,
         1                   NELEM, NXEQ, RESELN, STRESS, XYSO,OLDC,
         2 ALEN,QHIN,QHOUT,CCIN,QV,AWID,BWID,VSET,DENS,DELTD,DFZ,
         3        PUR, TBED)
C
C    THIS ROUTINE WRITES THE SIMULATION RESULTS TO THE RESULT FILE
C    (LUN 5) THAT HAS BEEN OPENED BY SERATRA
C
C     FORMAL PARAMETERS:
C        B      - BED CONCENTRATIONS
C        BDIV   - STANDARD BED LAYER THICKNESS
C        BED    - BED THICKNESS
C        C      - WATER CONCENTRATIONS
C        DELZ   - STANDARD ELEMENT THICKNESS
C        ELEV   - SEGMENT ELEVATION (DATUM ELEVATION)
C        NBED   - NUMBER OF BED LAYERS
C        NELEM  - NUMBER OF ELEMENTS
C        NXEQ   - CURRENT TIME STEP (I*4)
C        RESELN - WATER SURFACE ELEVATION
C        STRESS - SHEAR STRESS
C        XYSO   - THICKNESS OF THE TOP BED LAYER
C
C        CALLED BY: SERATRA.
C
          INCLUDE 'ELMSIZ.PRM'
C
          INTEGER*4 NXEQ
C
          DIMENSION B(MAXLEV,MAXCON+1),BAVG(MAXLEV), BEL(MAXLEV),
         1           C(MXELEM,MAXCON), CMASS(MXELEM), CVOLM(MXELEM),
         2           CTOTL(MXELEM), WELEV(MXELEM), D(MXELEM,MAXCON)
         3 ,QHIN(MXELEM),QHOUT(MXELEM),CCIN(MXELEM,MAXCON),
         4 QV(MXELEM),AWID(MXELEM),BWID(MXELEM),VSET(3),
         5 DENS(3),QCIN(MXELEM,MAXCON),QCOUT(MXELEM,MAXCON),
         6 CELAV(MXELEM,MAXCON),QVCEL(MXELEM,MAXCON),
         7 OLDC(MXELEM,MAXCON), DFZ(4), QVDIF(MXELEM,MAXCON),
         8        UTBED(MAXCON),TBED(MAXCON),BAL(MAXCON),DIF(MAXCON),
         9        QCELAV(MAXCON), TOTDIF(MAXCON)
          DATA EPSI/1.0E-30/
          NELMP1=NELEM+1
          NBEDP1 = NBED + 1
C
C         *** WATER CONCENTRATIONS ***
C
          J = NELEM + 1
          WRITE(5) NXEQ,J,NBED,ELEV,DELZ,BDIV,XYSO,STRESS
          REPEAT UNTIL (J .EQ. 0)
          .  CMASS(J) = C(J,4) + C(J,5) + C(J,6)
          .  SUM = C(J,1) + C(J,2) + C(J,3)
          .  WHEN (SUM .GT. 0.0) CVOLM(J) = CMASS(J) / SUM
          .  ELSE CVOLM(J) = 0.0
          .  CTOTL(J) = CMASS(J) + C(J,7)
          .  SELECT (J)
```

```
              .  .  (NELEM+1) WELEV(NELEM+1) = RESELN
              .  .  (OTHERWISE) WELEV(J) = WELEV(J+1) - DELZ
              .  ...FIN
              .  J = J - 1
              ...FIN
              DO(K=1,MAXCON)
              .  DO(I=1,NELEM+1)
              .  .  WHEN (K .LE. 3 .OR. K .EQ. 7) D(I,K)=C(I,K)
              .  .  ELSE
              .  .  .  D(I,K)=0.
              .  .  .  IF (C(I,K-3) .GT. 1.0E-10) D(I,K)=C(I,K)/C(I,K-3)
              .  .  ...FIN
              .  ...FIN
              ...FIN
              WRITE(5) (WELEV(J),(C(J,K),K=1,MAXCON),CMASS(J),CVOLM(J),
             1          CTOTL(J),J=NELEM+1,1,-1)
              WRITE(5) ((D(J,K),K=4,6),J=NELEM+1,1,-1)
        C
        C     *** BED CONCENTRATIONS ***
        C
              BELEV = (NBED-1) * BDIV + XYSO + ELEV
              J = NBED
              REPEAT UNTIL (J .EQ. 0)
              .  BAVG(J) = B(J,1)*B(J,4) + B(J,2)*B(J,5) + B(J,3)*B(J,6)
              .  SELECT (J)
              .  .  (NBED) BEL(NBED) = BELEV
              .  .  (NBED-1) BEL(NBED-1) = BELEV - XYSO
              .  .  (OTHERWISE) BEL(J) = BEL(J+1) - BDIV
              .  ...FIN
              .  J = J-1
              ...FIN
              WRITE(5) (BEL(J),(B(J,K),K=1,MAXCON-1),BAVG(J),J=NBED,1,-1)
        C    *** ELEMENT MASS AND CONVECTED MASS ***
              DO (J=1,MAXCON)
              .  OCELAV(J) = 0.
              .  CELAV(NELMP1,J)=0
              .  QCIN(NELMP1,J)=0
              .  QCOUT(NELMP1,J)=0
              .  DO(I =1,NELEM)
              .  .  VOL=AWID(I)*DELZ*ALEN
              .  .  XS=AWID(I)*DELZ
              .  .  VPXS=AWID(I)*ALEN
              .  .  OCMEAN = (OLDC(I,J) + OLDC(I+1,J))/2.
              .  .  CMEAN=(C(I,J)+C(I+1,J))/2.
              .  .  CELAV(I,J)=VOL*CMEAN
              .  .  QCIN(I,J)=QHIN(I)*(CCIN(I,J)+CCIN(I+1,J))/2.*DELTD
              .  .  QCOUT(I,J)=QHOUT(I)*DELTD*(CMEAN+(OLDC(I,J)+OLDC(I+1,J))/2.)/2.
              .  .  CELAV(NELMP1,J)=CELAV(NELMP1,J)+CELAV(I,J)
              .  .  QCIN(NELMP1,J)=QCIN(NELMP1,J)+QCIN(I,J)
              .  .  QCOUT(NELMP1,J)=QCOUT(NELMP1,J)+QCOUT(I,J)
              .  .  OCELAV(J) = OCELAV(J) + OCMEAN*VOL
              .  .  K=J
              .  .  IF(J .GT. 3) K=J-3
              .  .  WHEN (K .EQ. 4) WS=0.
              .  .  ELSE WS=VSET(K)
              .  .  WHEN(I .EQ. 1) QVCBTM=QV(1)*(C(1,J)+OLDC(1,J))/2.
```

```
              .  .  ELSE QVCBTM=(QV(I)-WS*BWID(I)*ALEN)*(C(I,J)+OLDC(I,J))/2.
              .  .  WHEN (I .EQ. NELEM)
              .  .  .  QVCTOP=QV(NELEM+1)*(C(NELEM+1,J)+OLDC(NELEM+1,J))/2.
              .  .  ...FIN
              .  .  ELSE
              .  .  .  QVCTOP=(QV(I+1)-WS*BWID(I+1)*ALEN)*(C(I+1,J)+OLDC(I+1,J))/2.
              .  .  ...FIN
              .  .  QVCEL(I,J)=(QVCBTM-QVCTOP)*DELTD
              .  .  QVDIF(I,J)= DFZ(K)*DELTD*AWID(I)*ALEN*
             1.  .  (C(I+1,J)-C(I,J)+OLDC(I+1,J)-OLDC(I,J))/(2.*DELZ)
              .  ...FIN
              ...FIN
C*********************************************************************
C      CHANGE 3/26/81
C
       VOLUME=0.0
       DO (J=1,NELEM) VOLUME=VOLUME+AWID(J)*DELZ*ALEN
       WRITE(5)VOLUME
C
C*********************************************************************
       WRITE(5)((CELAV(J,K),K=1,MAXCON),J=NELMP1,1,-1)
       WRITE(5)((QCIN(J,K),K=1,MAXCON),J=NELMP1,1,-1)
       WRITE(5)((QCOUT(J,K),K=1,MAXCON),J=NELMP1,1,-1)
       WRITE(5)((QVCEL(J,K),K=1,MAXCON),J=NELEM,1,-1)
       WRITE(5)((QVDIF(J,K),K=1,MAXCON),J=NELEM,1,-1)
C
       DO (J=1,MAXCON)
       .  BAL(J) = QCIN(NELMP1,J)-QCOUT(NELMP1,J)+OCELAV(J)
       .  DIF(J) = CELAV(NELMP1,J)-BAL(J)
       ...FIN
       TBAL=0.
       TDIF=0.
       DO (J=4,MAXCON)
       .  TBAL = TBAL + BAL(J)
       .  TDIF = TDIF + DIF(J)
       ...FIN
       WRITE(5) (BAL(J),J=1,MAXCON),TBAL
       WRITE(5) (DIF(J),J=1,MAXCON),TDIF
C
C    *** BED SEDIMENT AND CONTAMINANT (KG, PC)/ELEMENT ***
C
       DO (J=1,MAXCON)
       .  OTBED(J) = TBED(J)
       .  TBED(J) = 0.
       ...FIN
       VPXS=BWID(1)*ALEN
       DO (J=1,MAXCON-1)
       .  DO (I=1,NBED)
       .  .  WHEN (J .LT. 4)
       .  .  .  DNSITY=(1.0-POR)/(B(I,1)/DENS(1)+B(I,2)/DENS(2)+B(I,3)/DENS(3))
       .  .  .  DEL=BDIV
       .  .  .  IF(I.EQ. NBED) DEL=XY80
       .  .  .  VOL=DEL*VPXS
       .  .  .  CELAV(I,J)=B(I,J)*VOL*DNSITY
       .  .  ...FIN
       .  .  ELSE CELAV(I,J)=CELAV(I,J-3)*B(I,J)
```

```
         .  .  TBED(J) = TBED(J) + CELAV(I,J)
         .  ...FIN
         ...FIN
         WRITE(5) ((CELAV(J,K),K=1,MAXCON-1),J=NBED,1,-1)
         DO (J=4,MAXCON-1)
         .  TBED(MAXCON) = TBED(MAXCON) + TBED(J)
         ...FIN
         WRITE(5) (TBED(J),J=1,MAXCON)
         DO (J=1;MAXCON-1)
         .  TOTDIF(J) =  TBED(J) - OTBED(J) + DIF(J)
         ...FIN
         TOTDIF(MAXCON) = TBED(MAXCON) - OTBED(MAXCON) + TDIF
         WRITE(5) (TOTDIF(J),J=1,MAXCON)
         RETURN
         END
```

(FLECS VERSION 22.46)

```
        --------------------------------------
              SUBROUTINE SEDDAT(DECAY, DENS, DFZ, DIAM, DSHR, D50, ECHO,
             1                  ERODE, HLDERR, NUMERR, SCSHR, SORBK, VSET,
             2     DSORB)
      C
      C    THIS SUBROUTINE IS RESPONSIBLE FOR READING AND PROCESSING THE
      C    SEDIMENT CHARACTERISTICS.
      C
      C      FORMAL PARAMETERS:
      C        DECAY  - DECAY PARAMETERS
      C        DENS   - SPECIFIC WEIGHT
      C        DFZ    - VERTICAL DIFFUSION COEFFICIENTS
      C        DIAM   - PARTICLE DIAMETERS
      C        DSHR   - CRITICAL SHEAR STRESS VALUE FOR DEPOSITION
      C        D50    - MEDIAN BED SEDIMENT DIAMETER
      C        ECHO   - LINE PRINTER ECHO CONTROL VARIABLE (L*1)
      C        ERODE  - ERODABILITY
      C        HLDERR - HOLDING ARRAY FOR ERROR NUMBERS (BYTE)
      C        NUMERR - NUMBER OF INPUT ERRORS
      C        SCSHR  - CRITICAL SHEAR STRESS VALUE FOR SCOUR
      C        SORBK  - ADSORBTION VALUES
      C      DSORB  - DESORPTION VALUES
      C        VSET   - VERTICAL SETTLING VELOCITIES
      C
      C    CALLED BY:SERATRA
      C    CALLES: PUTERR
      C
              BYTE HLDERR(100)
      C
              LOGICAL*1 ECHO
      C
              DIMENSION DECAY(6),DENS(3),DFZ(4),DIAM(3),DSHR(3),ERODE(3),
             1          SCSHR(3),SORBK(9),DSORB(9),VSET(3)
      C
      C    ....PARTICLE SETTLINE VELOCITY (M/SEC)
      C
      C      COL.  1-10,...VSET(1)....SAND SETTLING VELOCITY
      C           11-20,...VSET(2)....SILT SETTLING VELOCITY
      C           21-30,...VSET(3)....CLAY SETTLING VELOCITY
      C
              READ(1,1) (VSET(I),I=1,3)
              IF(ECHO)
              .  WRITE(6,2)
              .  WRITE(6,12) (VSET(I),I=1,3)
              ...FIN
      C
      C    ....DENSITY (KG/M**3)
      C
      C      COL.  1-10,...DENS(1)....DENSITY OF SAND
      C           11-20,...DENS(2)....DENSITY OF SILT
      C           21-30,...DENS(3)....DENSITY OF CLAY
      C
              READ(1,1) (DENS(I),I=1,3)
              IF(ECHO)
```

```
            .  WRITE(6,3)
            .  WRITE(6,12) (DENS(I),I=1,3)
            ...FIN
            IF(DENS(1) .LE. 0.0) CALL PUTERR(17,NUMERR,HLDERR)
            IF(DENS(2) .LE. 0.0) CALL PUTERR(18,NUMERR,HLDERR)
            IF(DENS(3) .LE. 0.0) CALL PUTERR(19,NUMERR,HLDERR)
C
C     ....DIAMETER (METERS)
C
C       COL.  1-10....DIAM(1)....DIAMETER OF SAND
C            11-20....DIAM(2)....DIAMETER OF SILT
C            21-30....DIAM(3)....DIAMETER OF CLAY
C            31-40....D50........MEDIAN BED SEDIMENT DIAMETER
C
            READ(1,1) (DIAM(I),I=1,3),D50
            IF(ECHO)
            .  WRITE(6,4)
            .  WRITE(6,12) (DIAM(I),I=1,3)
            .  WRITE(6,13) D50
            ...FIN
            IF(DIAM(1) .LE. 0.0) CALL PUTERR(20,NUMERR,HLDERR)
            IF(DIAM(2) .LE. 0.0) CALL PUTERR(21,NUMERR,HLDERR)
            IF(DIAM(3) .LE. 0.0) CALL PUTERR(22,NUMERR,HLDERR)
C
C     ....CRITICAL SHEAR STRESS FOR SCOUR (KG/M**2)
C
C       COL.  1-10....SCSHR(1)....CRITICAL SHEAR STRESS FOR SAND
C            11-20....SCSHR(2)....CRITICAL SHEAR STRESS FOR SILT
C            21-30....SCSHR(3)....CRITICAL SHEAR STRESS FOR CLAY
C
            READ(1,1) (SCSHR(I),I=1,3)
            IF(ECHO)
            .  WRITE(6,5)
            .  WRITE(6,12) (SCSHR(I),I=1,3)
            ...FIN
            IF(SCSHR(1) .LE. 0.0) CALL PUTERR(26,NUMERR,HLDERR)
            IF(SCSHR(2) .LE. 0.0) CALL PUTERR(27,NUMERR,HLDERR)
            IF(SCSHR(3) .LE. 0.0) CALL PUTERR(28,NUMERR,HLDERR)
C
C     ....CRITICAL SHEAR STRESS FOR DEPOSITION (KG/M**2)
C
C       COL.  1-10....DSHR(1)....CRITICAL SHEAR STRESS FOR SAND
C            11-20....DSHR(2)....CRITICAL SHEAR STRESS FOR SILT
C            21-30....DSHR(3)....CRITICAL SHEAR STRESS FOR CLAY
C
            READ(1,1) (DSHR(I),I=1,3)
            IF(ECHO)
            .  WRITE(6,6)
            .  WRITE(6,12) (DSHR(I),I=1,3)
            ...FIN
            IF(DSHR(1) .LE. 0.0) CALL PUTERR(29,NUMERR,HLDERR)
            IF(DSHR(2) .LE. 0.0) CALL PUTERR(30,NUMERR,HLDERR)
            IF(DSHR(3) .LE. 0.0) CALL PUTERR(31,NUMERR,HLDERR)
C
C     ....ERODABILITY (KG/M**2/SEC)
C
```

```
C      COL.  1-10....ERODE(1)....ERODABILITY OF SAND
C           11-20....ERODE(2)....ERODABILITY OF SILT
C           21-30....ERODE(3)....ERODABILITY OF CLAY
C
       READ(1,1) (ERODE(I),I=1,3)
       IF(ECHO)
       .  WRITE(6,7)
       .  WRITE(6,12) (ERODE(I),I=1,3)
       ...FIN
       IF(ERODE(1) .LE. 0.0) CALL PUTERR(32,NUMERR,HLDERR)
       IF(ERODE(2) .LE. 0.0) CALL PUTERR(33,NUMERR,HLDERR)
       IF(ERODE(3) .LE. 0.0) CALL PUTERR(34,NUMERR,HLDERR)
C
C   ....VERTICAL DIFFUSION COEFFICIENTS (M**2/SEC)
C
C      COL.  1-10....DFZ(1)....COEFFICIENT FOR SAND
C           11-20....DFZ(2)....COEFFICIENT FOR SILT
C           21-30....DFZ(3)....COEFFICIENT FOR CLAY
C           31-40....DFZ(4)....COEFFICIENT FOR DISSOLVED CONTAMINANT
C
       READ(1,1) (DFZ(I),I=1,4)
       IF(ECHO)
       .  WRITE(6,8)
       .  WRITE(6,12) (DFZ(I),I=1,3)
       .  WRITE(6,14) DFZ(4)
       ...FIN
C
C   ....ADSORBTION VALUES (2 CARDS)
C
C      CARD #1
C        COL.  1-10....SORBK(1)....KD VALUE WITH SAND (M**3/KG)
C             11-20....SORBK(2)....KD VALUE WITH SILT (M**3/KG)
C             21-30....SORBK(3)....KD VALUE WITH CLAY (M**3/KG)
C             31-40....SORBK(4)....SUSPENDED SAND MASS TRANSFER RATE (1/S
C             41-50....SORBK(5)....SUSPENDED SILT MASS TRANSFER RATE (1/S
C             51-60....SORBK(6)....SUSPENDED CLAY MASS TRANSFER RATE (1/S
C             61-70....SORBK(7)....BED SAND MASS TRANSFER RATE (1/SEC)
C             71-80....SORBK(8)....BED SILT MASS TRANSFER RATE (1/SEC)
C      CARD #2
C        COL.  1-10....SORBK(9)....BED CLAY MASS TRANSFER RATE (1/SEC)
C
       READ(1,1) (SORBK(I),I=1,9)
C
C   ....DESORPTION VALUES (2 CARDS)
C
C      CARD #1
C        COL.  1-10....DSORB(1)....KD VALUE WITH SAND (M**3/KG)
C             11-20....DSORB(2)....KD VALUE WITH SILT (M**3/KG)
C             21-30....DSORB(3)....KD VALUE WITH CLAY (M**3/KG)
C             31-40....DSORB(4)....SUSPENDED SAND MASS TRANSFER RATE (1/S
C             41-50....DSORB(5)....SUSPENDED SILT MASS TRANSFER RATE (1/S
C             51-60....DSORB(6)....SUSPENDED CLAY MASS TRANSFER RATE (1/S
C             61-70....DSORB(7)....BED SAND MASS TRANSFER RATE (1/SEC)
C             71-80....DSORB(8)....BED SILT MASS TRANSFER RATE (1/SEC)
C      CARD #2
C        COL.  1-10....DSORB(9)....BED CLAY MASS TRANSFER RATE (1/SEC)
```

```
      C
                READ(1,1) (DSORH(I),I=1,9)
                IF(ECHO)
                .  WRITE(6,9)
                .  WRITE(6,12) (SORBK(I),I=1,3)
                .  WRITE(6,15) (SORBK(I),I=4,9)
                .  WRITE(6,16)
                .  WRITE(6,12) (DSORB(I),I=1,3)
                .  WRITE(6,15) (DSORB(I),I=4,9)
                ...FIN
      C
      C        ....DECAY PARAMETERS (2 CARDS)
      C
      C         CARD #1
      C           COL.  1-10....DECAY(1)....RADIONUCLIDE DECAY (1/SEC)
      C                11-20....DECAY(2)....TOTAL DECAY, SUM OF ALL DECAY EXCEPT
      C                                     FOR RADIONUCLIDE DECAY, IF GIVEN, THE
      C                                     REMAINING PARAMETERS ARE NOT TO BE
      C                                     SUPPLIED, PESTICIDE ONLY (1/SEC)
      C                21-30....DECAY(6)....VOLATIZATION DEGRADATION RATE,
      C                                     PESTICIDE ONLY (1/SEC)
      C                31-40....PH..........DEGREE OF ACIDITY OF ALKALINITY
      C                41-50....AKA.........SECOND ORDER ACID RATE CONSTANT
      C                                     FOR HYDROLYSIS
      C                51-60....AKB.........SECOND ORDER BASE RATE CONSTANT
      C                                     FOR HYDROLYSIS
      C                61-70....AKN.........SECOND ORDER RATE CONSTANT OF NEUTROL
      C                                     REACTION WITH WATER
      C                71-80....AKOX........SECOND ORDER RATE CONSTANT OF FREE
      C                                     RADICAL OXYGEN FOR OXIDATION
      C         CARD #2
      C           COL   1-10....RO2.........CONCENTRATION OF FREE RADICAL OXYGEN
      C                11-20....AKBIO.......SECOND ORDER RATE CONSTANT
      C                                     BIODEGRADATION
      C                21-30....BIOMAS......BIOMASS PER UNIT VOLUME
      C
                READ(1,1) DECAY(1),DECAY(2),DECAY(6),PH,AKA,AKB,AKN,AKOX,RO2,
               1         AKBIO,BIOMAS
                DO (I=3,5) DECAY(I) = 0.0
                WHEN (DECAY(2) .NE. 0.0)
                .  IF(ECHO) WRITE(6,10) DECAY(1),DECAY(2)
                ...FIN
                ELSE
      C         .  *** COMPUTE: DECAY(3) - CHEMICAL DEGRADATION DUE TO HYDROLYSIS
      C         .               DECAY(4) - CHEMICAL DEGRADATION DUE TO OXIDATION
      C         .               DECAY(5) - BIODEGRADATION  ***
                .  DECAY(3) = 10.0**(PH-14.0)*AKB + 10.0**(-PH)*AKA + AKN
                .  DECAY(4) = AKOX * RO2
                .  DECAY(5) = AKBIO * BIOMAS
                .  IF(ECHO)
                .  .  WRITE(6,11) DECAY(1),PH,AKA,AKB,AKN,DECAY(3),AKOX,RO2,DECAY(4),
               1.  .              AKBIO,BIOMAS,DECAY(5),DECAY(6)
                .  ...FIN
                ...FIN
      C
                RETURN
```

```
C
1     FORMAT(8F10.0)
2     FORMAT(1H0,13X,'PARTICLE SETTLING VELOCITY (M/SEC)')
3     FORMAT(1H0,13X,'DENSITY (KG/M**3)')
4     FORMAT(1H0,13X,'DIAMETER (METERS)')
5     FORMAT(1H0,13X,'CRITICAL SHEAR STRESS FOR SCOUR (KG/M**2)')
6     FORMAT(1H0,13X,'CRITICAL SHEAR STRESS FOR DEPOSITION (KG/M**2)')
7     FORMAT(1H0,13X,'ERODABILITY (KG/M**2/SEC)')
8     FORMAT(1H0,13X,'VERTICAL DIFFUSION COEFFICIENTS (M**2/SEC)')
9     FORMAT(1H0,13X,'ADSORPTION KD VALUES')
10    FORMAT(1H0,13X,'DECAY PARAMETERS'/
     1  14X,1PE12.5,'...RADIONUCLIDE DECAY (1/SEC)'/
     2  14X,1PE12.5,'...TOTAL DECAY (EXCEPT RADIONUCLDE) - (1/SEC)')
11    FORMAT(1H0,13X,'DECAY PARAMETERS'/
     1  14X,1PE12.5,'...RADIONUCLIDE DECAY (1/SEC)'/
     2  14X,1PE12.5,'...PH - DEGREE OF ACIDITY OR ALKALINITY (PH)'/
     3  14X,1PE12.5,'...SECOND ORDER ACID RATE CONSTANT FOR HYDROLYSIS'
     4                     ' (AKA)'/
     5  14X,1PE12.5,'...SECOND ORDER BASE RATE CONSTANT FOR HYDROLYSIS'
     6                     ' (AKB)'/
     7  14X,1PE12.5,'...SECOND ORDER RATE CONSTANT OF NEUTROL REACTION'
     8                     ' WITH WATER (AKN)'/
     9  14X,1PE12.5,'...CHEMICAL DEGRADATION DUE TO HYDROLYSIS'/
     1  14X,1PE12.5,'...SECOND ORDER RATE CONSTANT OF FREE RADICAL'
     2                     ' OXYGEN (AKOX)'/
     3  14X,1PE12.5,'...CONCENTRATION OF FREE RADICAL OXYGEN (RO2)'/
     4  14X,1PE12.5,'...CHEMICAL DEGRADATION DUE TO OXIDATION'/
     5  14X,1PE12.5,'...SECOND ORDER RATE CONSTANT FOR BIODEGRADATION'
     6                     ' (AKBIO)'/
     7  14X,1PE12.5,'...BIOMASS PER UNIT VOLUME (BIOMAS)'/
     8  14X,1PE12.5,'...BIODEGRADATION'/
     9  14X,1PE12.5,'...VOLATILIZATION')
12    FORMAT(14X,1PE12.5,'...SAND'/
     1  14X,1PE12.5,'...SILT'/
     2  14X,1PE12.5,'...CLAY')
13    FORMAT(14X,1PE12.5,'...MEDIAN BED SEDIMENT DIAMETER')
14    FORMAT(14X,1PE12.5,'...DISSOLVED CONTAMINANT')
15    FORMAT(1H0,13X,'MASS TRANSFER RATES (1/SEC)'/
     1  14X,1PE12.5,'...SUSPENDED SAND'/
     2  14X,1PE12.5,'...SUSPENDED SILT'/
     3  14X,1PE12.5,'...SUSPENDED CLAY'/
     4  14X,1PE12.5,'...SAND ATTACHED TO THE BED'/
     5  14X,1PE12.5,'...SILT ATTACHED TO THE BED'/
     6  14X,1PE12.5,'...CLAY ATTACHED TO THE BED')
16    FORMAT(1H0,13X,'DESORPTION KD VALUES')
C
      END
```

(FLECS VERSION 22.46)

```
        ----------------------------------------

            SUBROUTINE SEDIME(ABAR, ALEN, CCIN, DELTD, I, J, NELEM,
           1                  QHIN, QHOUT, QV, SD, SR, ALFA, BETA, VEL1,
           2                  VEL2, VSET,  DELZ, BWID, AWID,DEPTH,
           3                  BETA1, BETA2)
      C
      C   THIS ROUTINE CALCULATES COEFFICIENTS OF CONVECTION, DIFFUSION,
      C   DECAY AND SOURCE TERMS IN THE SEDIMENT TRANSPORT CONVECTION
      C   -DIFFUSION EQUATION
      C
      C     INPUT PARAMETERS:
      C       ABAR   - AVERAGE AREA
      C       CCIN   - CONCENTRATION OF INFLOW
      C       DELTD  - TIME STEP IN DAYS
      C       DEPTH  - DEPTH OF RIVER SEGMENT
      C       I      - ELEMENT INDEX
      C       J      - PARAMETER INDEX
      C       NELEM  - NUMBER OF ELEMENTS
      C       QHIN   - INFLOW DISCHARGE
      C       QHOUT  - OUTFLOW DISCHARGE
      C       QV     - VERTICAL DISCHARGE
      C       SD     - SEDIMENT DEPOSITION RATE, (KG/M**3/DAY)
      C       SR     - SEDIMENT EROSION RATE, (KG/M**3/DAY)
      C     OUTPUT PARAMETERS:
      C       ALFA   - DECAY TERM, (1/DAY)
      C       BETA   - SOURCE OR SINK TERM, (KG/M**3/DAY)
      C       BETA1  - INFLUENT SOURCE TERM FOR THE I-TH NODE, (KG/M**3/DAY)
      C          BETA2  - INFLUENT SOURCE TERM FOR THE I+1-TH NODE, (KG/M**3/D
      C       VEL1   - FIRST CONVECTIVE TERM, (M/DAY)
      C       VEL2   - SECOND CONVECTIVE TERM, (M/DAY)
      C
      C     CALLED BY TRANSP.
      C
            INCLUDE 'SY:ELMSIZ.PRM'
      C
            DIMENSION ABAR(MXELEM), AREA(MXELEM), CCIN(MXELEM,MAXCON),
           1          QHIN(MXELEM), QHOUT(MXELEM), QV(MXELEM), SD(MXELEM,6),
           2          SR(6), VSET(3), BWID(MXELEM), AWID(MXELEM)
      C
      C   CONVECTIVE TERM WITH CORRECTION FOR A CONTINUOUS SETTLING FLUX
      C
            AQ = QV(I)
            VEL1=(AQ-VSET(J)*BWID(I)*ALEN)/ABAR(I)
            AQ = QV(I+1)
            VEL2=(AQ-VSET(J)*BWID(I+1)*ALEN)/ABAR(I)
      C
      C   DECAY TERM
      C
            ALFA = QHOUT(I) / (ABAR(I) * DELZ)
      C
      C   SOURCE OR SINK TERM
      C
            BETA=SR(J) - SD(I,J)
            BETA1=QHIN(I)/(ABAR(I)*DELZ)*(CCIN(I,J)/3.+CCIN(I+1,J)/6.)
```

```
 00054          BETA2=QMIN(I)/(ABAR(I)*DELZ)*(CCIN(I,J)/6.+CCIN(I+1,J)/3.)
 00055          RETURN
 00056          END

        (FLECS VERSION 22.46)

        ----------------------------------------
```

```
      ----------------------------------------
C                                 [106,1]SERATRA.FLX
C***********************************************************************
C       VERSION: DRIED CHANNEL OPTION
C                WITH SELECTION FOR IMPLICIT SETTLING VELOCITY SCHEME
C                   OR SUPERPOSITION OF SETTLING VELOCITY SCHEME
C***********************************************************************
C
C
C      THIS COMPUTER PROGRAM, SERATRA, IS AN UNSTEADY, TWO-
C      DIMENSIONAL (LONGITUDINAL AND VERTICAL) MODEL TO SIMULATE
C      SEDIMENT-CONTAMINANT TRANSPORT IN RIVERS AND RIVER-RUN
C      RESERVOIRS.
C
C      THE MODEL HAS GENERAL CONVECTION-DIFFUSION EQUATIONS
C      WITH DECAY AND SINK/SOURCE TERMS WITH APPROPRIATE BOUNDARY
C      CONDITIONS.
C
C      SERATRA UTILIZES THE FINITE ELEMENT COMPUTATION METHOD WITH
C      THE GALERKIN WEIGHTED RESIDUAL TECHNIQUE.
C
C      THE FOLLOWING REPORTS DESCRIBE SERATRA MODEL FORMULATION, USER'S
C      MANUAL AND SOME MODEL RESULTS:
C
C**
C 1 ONISHI,Y.,P.A.JOHANSON,R.G.BACA,AND E.L.HILTY.1976.
C   "STUDIES OF COLUMBIA RIVER WATER QUALITY--DEVELOPMENT OF
C   MATHEMATICAL MODELS FOR SEDIMENT AND RADIONUCLIDE TRANSPORT
C   ANALYSIS." BNWL-B-452. BATTELLE, PACIFIC NORTHWEST
C   LABORATORIES, RICHLAND, WA.
C**
C 2 ONISHI,Y. 1977. "FINITE ELEMENT MODELS FOR SEDIMENT AND CONTAMINANT
C   TRANSPORT IN SURFACE WATERS--TRANSPORT OF SEDIMENT AND RADIONUCLIDES
C   IN THE CLINCH RIVER." BNWL-2227. BATTELLE, PACIFIC NORTHWEST
C   LABORATORIES, RICHLAND, WA.
C**
C 3 ONISHI,Y. 1977. "MATHEMATICAL SIMULATION OF SEDIMENT AND RADIO-
C   NUCLIDE TRANSPORT IN THE COLUMBIA RIVER." BNWL-2228. BATTELLE,
C   PACIFIC NORTHWEST LABORATORIES, RICHLAND, WA.
C**
C 4 ONISHI,Y., D.L. SCHREIBER AND R.B. CODELL. 1979. "MATHEMATICAL
C   SIMULATION OF SEDIMENT AND RADIONUCLIDE TRANSPORT IN THE CLINCH
C   RIVER, TENNESSEE." PROCEEDINGS OF ACS/CSJ CHEMICAL CONGRESS,
C   HONOLULU, HAWAII, APRIL 1-6, 1979. "CONTAMINANTS AND SEDIMENTS",
C   R.A. BAKER (ED.), ANN ARBOR SCIENCE PUBLISHERS, INC., ANN ARBOR, MI.
C**
C 5 ONISHI,Y., S.M. BROWN, A.R. OLSEN, M.A. PARKHURST, S.E. WISE, AND
C   W.H. WALTERS. 1979. "METHODOLOGY FOR OVERLAND AND INSTREAM MIGRATION
C   AND RISK ASSESSMENT OF PESTICIDES;" BATTELLE, PACIFIC NORTHWEST
C   LABORATORIES, RICHLAND, WA.
C**
C 6 ONISHI,Y. AND S.E. WISE. 1979. "MATHEMATICAL MODEL, SERATRA, FOR
C   SEDIMENT AND CONTAMINANT TRANSPORT IN RIVERS AND ITS APPLICATION TO
C   PESTICIDE TRANSPORT IN FOUR MILE AND WOLF CREEKS IN IOWA." BATTELLE,
```

```
      C   PACIFIC NORTHWEST LABORATORIES, RICHLAND, WA.
      C**
      C 7 ONISHI,Y. AND S.E. WISE. 1979. "USER'S MANUAL FOR THE INSTREAM
      C   SEDIMENT-CONTAMINANT TRANSPORT MODEL, SERATRA," BATTELLE, PACIFIC
      C   NORTHWEST LABORATORIES, RICHLAND, WA.
      C
      C
      C   CALLS: BEDDAT, COLAPS, DIMDAT, FCODE, HYDDAT, HYDFLO, ICFLO, INIDAT,
      C          RDSFLO, RPTERR, SAVEIT, SEDDAT, STRTUP, TRANSP, TRBDAT,
      C          TRBFLO, UPSDAT, WTRDAT, PHOINP
      C
            INCLUDE 'ELMSIZ.PRM'
      C
            BYTE FNAME(29),BASE(5),FTYPE(3),DEV(3),GUIC(3),UUIC(3),
           1     HLDERR(100),NOW(8)
      C
            LOGICAL*1 ECHO,ANALYS,NEWTRB,NEWIC,FERROR,NEWQI,RIVER,
           1          ECHO2,ECHO3,ECHO4,ECHO5,ECHO6,ECHO7,ECHO8,ECHO9,ECHO10,
           2          SAVECH
      C
            INTEGER*2 OUTFLO,TRBOPT
      C
            INTEGER*4 ETIME,ENDHYD,ENDIC,ENDTRB,SIMLEN,DU42,
           1           NSTEPS,NXEQ,NFRST,DUM1,NSP,SECYR,JULSEC
      C
      C
            DIMENSION AWID(MXELEM), CLAST(MXELEM,MAXCON), CTRB(MXELEM,MAXCON),
           1            PTQAVG(MXELEM),ELMVOL(MXELEM),
           2QAVG(MXELEM),CTHIB(MAXCON),PCOEF(4),CNODE(MXELEM,MAXCON),
           3           EL(MXELEM), XSAREA(MXELEM), BWID(MXELEM), TMASS(MAXCON),
           4 IELM(MXELEM),UWID(MXELEM),UEL(MXELEM),ABAR(MXELEM),PWID(MXELEM),
           5 PXSAR(MXELEM), OLDC(MXELEM,MAXCON), CDUMMY(MXELEM),
           6 TBED(MAXCON),  JSEG(5), SAVECH(10)
      C
            INCLUDE 'TRANS.COM'
      C
            DATA SECDAY/86400./
            DATA SECYR /31536000/
            DATA ECHO/.FALSE./
            DATA FERROR/.FALSE./
            DATA ZERO/1.0E-10/
            DATA SAVECH/.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE
           1.,.FALSE.,.FALSE.,.FALSE./
      C
              IS1=1
              IS2=0
            CALL STRTUP(BASE, DEV, ECHO, FNAME, FTYPE, GUIC, INFLO, ISTRT,
           1              NFRST, OUTFLO, SMETH, UUIC,  SAVECH, JSEG,
           2      NSTEPS,NSEG,ITPRT,ANALYS,DELTH,ANALMT,DEPMIN,SIMLEN,
           3      PELEV)
      C
      C     *** READ INPUT DATA FOR BEGINNING SEGMENT ***
            NUMERR = 0
      C
      C     THIS IS THE RESTART OPTION.
      C
```

```
               WHEN(ISTRT.NE.1)DELTD=DELTH/SECDAY
      C
               ELSE
               .  CALL INIDAT(ANALMT, ANALYS, DELTH, ECHO, HLDERR, ITPRT,
              1.               NSEG, NSTEPS, NUMERR, SIMLEN, DEPMIN)
               .  DELTD=DELTH/SECDAY
               .  IF(ECHO)
               .  .  CALL TIME(NOW)
               .  .  WRITE(6,3) ISTRT, NOW
               .  ...FIN
               .  CALL DIMDAT(ALEN, AREA, BDIV, BED, DLZSAV, ECHO, ELEV, HLDERR,
              1.               ISTRT, NBED, NELEM, NUMERR, PELEV, POR, RIVER,
              2.               XYSO,  EL)
               .  CALL SEDDAT(DECAY, DENS, DFZ, DIAM, DSHR, D50, ECHO, ERODE,
              1.               HLDERR, NUMERR, SCSHR, SORBK, VSET, DSORB)
               .  CALL PHOINP(ECHO,JULIAN,KAY1,KAY2,PCOEF)
               .  CALL BEDDAT(B, ECHO, NBED,AREA,BDIV,DENS,POR,XYSO,TBED)
      C        .
      C        .  READS INITIAL CONDITIONS FOR SEDIMENT AND CONTAMINANT
      C        .
               .  CALL WTRDAT(C, ECHO, NELEM)
      C        .
      C        .  READS UPSTREAM BOUNDARY CONDITIONS FOR FIRST SEGMENT
      C        .
               .  CALL UPSDAT(ECHO, HLDERR, NUMERR, SIMLEN, UWID, UEL)
               .  CALL TRBDAT(ECHO, HLDERR, NELEM, NTRIBS, NUMERR,
              1.               SIMLEN, TRBOPT)
               .  CALL HYDDAT(ALEN, AREA, DELTH, DELZ, D50, ECHO, HLDERR, NSETS,
              1.               NUMERR, SIMLEN,DEPMIN, DLZSAV, EL)
      C        .
               .  IF (NUMERR .GT. 0)
               .  .  CALL RPTERR(NUMERR, HLDERR, FERROR)
               .  .  IF (FERROR) REPORT-FATAL-ERROR-AND-STOP
               .  ...FIN
      C        .
               ...FIN
      C
               IF (ANALYS)
               .  OPEN(UNIT=8,NAME='SY:TIMSERIES.DAT',TYPE='NEW',
              1.        FORM='UNFORMATTED')
               ...FIN
      C**********************************
      C        SEGMENT LOOP                    *
      C**********************************
               DO (ISEG=ISTRT,NSEG)
               .  CALL DIAG(ECHO2, ECHO3, ECHO4, ECHO5, ECHO6, ECHO7, ECHO8,
              1.            ECHO9, ECHO10, ISEG, JSEG, SAVECH)
      3        .  FORMAT('1',100X,'SEGMENT NO.',I3,1X,8A1)
               .  RESET-DATA-TIME-CONTROLS
               .  CAVGMX=0.0
               .  JULSEC = JULIAN * SECDAY
      C        .
               .  UNLESS (ISEG .EQ. 1)
               .  .  IF (ECHO)
               .  .  .  CALL TIME(NOW)
               .  .  .  WRITE (6,3) ISEG,NOW
```

```
          .  .  ...FIN
          .  .  FERROR = .FALSE.
          .  .  NUMERR = 0
          .  .  CALL DIMDAT(ALEN, AREA, BDIV, BED, DLZ3AV, ECHO, ELEV, HLDERR,
         1.  .                  ISEG, NBED, NELEM, NUMERR, PELEV, POR, RIVER,
         2.  .                  XYSO,  FL)
          .  .  CALL SEDDAT(DECAY, DENS, DFZ, DIAM, DSHR, D50, ECHO, ERODE,
         1.  .                  HLDERR, NUMERR, SCSHR, SORBK, VSET, DSORB)
          .  .  CALL BEDDAT(B, ECHO, NBED,AREA,BDIV,DENS,POR,XYSO,THED)
 C        .  .
 C        .  .  READS INITIAL CONDITIONS FOR SEDIMENT AND CONTAMINANT
 C        .  .
          .  .  CALL WTRDAT(C, ECHO, NELEM)
          .  .  CALL TRBDAT(ECHO, HLDERR, NELEM, NTRIBS, NUMERR,
         1.  .                  SIMLEN, TRBOPT)
          .  .  CALL HYDDAT(ALEN, AREA, DELTH, DELZ, D50, ECHO, HLDERR, NSETS,
         1.  .                  NUMERR, SIMLEN, DEPMIN, DLZSAV, EL)
 C        .  .
          .  .  IF (NUMERR .GT. 0)
          .  .  .  CALL RPTERR(NUMERR, HLDERR, FERROR)
          .  .  .  IF (FERROR) REPORT-FATAL-ERROR-AND-STOP
          .  .  ...FIN
          .  ...FIN
          .  RESELN=ELEV+BED
 C        .
 C        .  *** READ THE HYDROLOGICAL DATA ***
          .  READ(4) DUM1,NELEM,DELZ,QI,QO,VOL,VEL,AWID,AREA,TEMPR,XSAREA,IELM,
         1.  DEPTH,BWID,ABAR,HRAD, CROSEC
          .  REWIND 4
 C        .
 C        .  PROCESSING OF INITIAL DATA
 C        .
          .  IF (NELEM .GT. 0)
          .  .  DO (K=1,MAXCON)
 C        .  .  .  *** NELEM VALUES OF CLAST MUST BE CALUCLATED FROM NELEM+1
 C        .  .  .       VALUES OF C ***
          .  .  .  IF (K .LE. 3 .OR. K .EQ. 7)
 C        .  .  .  .
 C        .  .  .  .  PROVIDES A CONSISTENT INITIAL CONDITION FOR THE DELZ DEFINED
 C        .  .  .  .  IN HYDDAT - BOTH NODAL VALUES (CNODE=C), AND ELEMENT
 C        .  .  .  .  AVERAGES (CLAST=COLD)
 C        .  .  .  .
          .  .  .  .  CALL COLLAP(ALEN,AREA,C,DELZ,EL,IELM,K,NELEM,CLAST,TMASS,AWID,
         1.  .  .  .      VSET,DFZ,CNODE,XSAREA)
          .  .  .  ...FIN
          .  .  ...FIN
 C        .  .
 C        .  .  DIAGNOSTIC WRITES
 C        .  .
          .  .  IF (ECHO5)
          .  .  .  WRITE(6,1500)
          .  .  .  WRITE(6,1510)
          .  .  .  WRITE(6,1520) (I,AREA(I),EL(I),ABAR(I),AWID(I),BWID(I),
         1.  .  .                 XSAREA(I),TMASS(I), I=1,NELEM+1)
          .  .  .  WRITE(6,1530)
          .  .  .  WRITE(6,1520) (I,(C(I,J),J=1,MAXCON),I=1,MXELEM)
```

```
                   .   .   .   WRITE(6,1540)
                   .   .   .   WRITE(6,1520) (I,(CNODE(I,J),J=1,MAXCON),I=1,NELEM+1)
                   .   .   .   WRITE(6,1550)
                   .   .   .   WRITE(6,1520) (I,(CLAST(I,J),J=1,MAXCON),I=1,NELEM)
        1500 .     .   .   .   FORMAT('0IMMEDIATELY FOLLOWING COLLAP, THE INITIAL CONDITION')
        1510 .     .   .   .   FORMAT('0ELEMENT/NODE',5X,'AREA',10X,'EL',12X,'ABAR',
                1. .   .   .        11X,'AWID',11X,'HWID',10X,'XSAREA',10X,'TMASS')
        1520 .     .   .   .   FORMAT(2X,I5,3X,1P7E15.4)
        1530 .     .   .   .   FORMAT('0NODAL CONCENTRATIONS PRIOR TO COLLAP')
        1540 .     .   .   .   FORMAT('0NODAL CONCENTRATIONS FOLLOWING COLLAP')
        1550 .     .   .   .   FORMAT('0ELEMENT AVERAGE CONCENTRATION FOLLOWING COLLAP')
                   .   .   ...FIN
                   .   .   DO (I=1,NELEM)
                   .   .   .   DO (J=1,MAXCON)
                   .   .   .   .   COLD(I,J)=CLAST(I,J)
                   .   .   .   .   C(I,J)=CNODE(I,J)
                   .   .   .   ...FIN
                   .   .   ...FIN
                   .   .   DO (J=1,MAXCON) C(NELEM+1,J)=CNODE(NELEM+1,J)
                   .   .   DO (J=1,MAXCON) COLD(NELEM+1,J)=0.0
                   .   .   IF(NELEM+1.LT.MXELEM)
                   .   .   .   DO (I=NELEM+2,MXELEM)
                   .   .   .   .   DO (J=1,MAXCON)
                   .   .   .   .   .   COLD(I,J) = 0.0
                   .   .   .   .   .   C(I,J) = 0.0
                   .   .   .   .   ...FIN
                   .   .   .   ...FIN
                   .   .   ...FIN
                   .   ...FIN
                   .   CALL FCODE(FNAME, BASE, ISEG, FTYPE, DEV, GUIC, UUIC)
                   .   OPEN(UNIT=5,NAME=FNAME,TYPE='NEW',FORM='UNFORMATTED')
                   .   OPEN(UNIT=9,NAME='RSTRT.FIL',TYPE='NEW',FORM='UNFORMATTED')
                   .   WRITE(5) ISEG
      C            .
      C            .   *** CONVERT INPUT VALUES TO THOSE UNITS USED BY MODEL ***
                   .   DO (J=1,3) VSET(J)=VSET(J) * SECDAY
                   .   DO (J=1,4) DFZ(J)=DFZ(J) * SECDAY
                   .   DO (I=1,6) DECAY(I) = DECAY(I) * SECDAY
                   .   DO (I=4,9) SORBK(I) = SORBK(I) * SECDAY
                   .   DO (I=4,9) DSORB(I) = DSORB(I) * SECDAY
      C*********************************************
      C            .               TIME STEP LOOP               *
      C*********************************************
      C            .
                   .   NXEQ = NFRST
                   .   IF (NXEQ.EQ.1)
                   .   .   PTDPTH = DEPTH
                   .   .   PTDELZ = DELZ
                   .   .   NELMPT = NELEM
                   .   ...FIN
                   .   ETIME = NXEQ - 1
                   .   ETIME = ETIME * DELTH
                   .   UNTIL (NXEQ .GT. NSTEPS)
                   .   .   IF (IS1 .EQ. 1) WRITE(11,4) ISEG,NXEQ
      4            .   .   FORMAT(' SEGMENT #',I3,' TIME STEP',I10)
                   .   .   ETIME = ETIME + DELTH
```

```
C       .  .  *** UPDATE THE FLOW AND CONCENTRATION ARRAYS ***
           .  .  CALL HYDFLO(ALEN, AREA, AWID, DELZ, DEPTH, D50, ELEV,
      1.   .                ENDHYD, ETIME, FERROR, HRAD, NELEM, NEWQI, PELEV,
      2.   .                QHIN, QHOUT, QV, RIVER, SLOPE, STRESS, TEMPR,
      3.   .                VEL, VOL, DEPMIN, XSAREA, BWID, ABAR, QI, CROSEC)
C          .  .
           .  .  IF (FERROR) REPORT-FATAL-ERROR-AND-STOP
C          .  .
           .  .  IF (NTRIBS .GT. 0)
           .  .  .  CALL TRBFLO(CTRB, CTRIB, ENDTRB, ETIME, FERROR, DEPTH, NELEM,
      1.   .  .  .                NEWQI, NEWTRB, QHIN, TRBOPT, DEPMIN)
           .  .  .  IF(ECHO3)
           .  .  .  .  WRITE(6,1100)
 1100      .  .  .  .  FORMAT('0AFTER TRBFLO, MASS FLUX VALUES OF CTRB')
           .  .  .  .  WRITE(6,1110)
 1110      .  .  .  .  FORMAT('0ELEMENT NO.',2X,3(3X,'CONC. OF',4X),3(1X,'CONC.'
      1.   .  .  .  .   'ASSOC.',2X),2X,'CONCAMINANT'/13X,'SUSPENDED SAND',1X,
      2.   .  .  .  .   'SUSPENDED SILT',1X,'SUSPENDED CLAY', 3X,'WITH SAND', 6X,
      3.   .  .  .  .   'WITH SILT', 6X,'WITH CLAY', 4X,'DISSOLVED CONC.')
           .  .  .  .  WRITE(6,1020)(I,(CTRB(I,K),K=1,MAXCON),I=1,NELEM)
           .  .  .  ...FIN
           .  .  .  IF (FERROR) REPORT-FATAL-ERROR-AND-STOP
           .  .  ...FIN
C          .  .
           .  .  CALL ICFLO(CCIN, DEPTH, DELZ, D50, ENDIC, ETIME, FERROR, INFLO,
      1.   .  ISEG,NELEM,NEWIC,QHIN,QI,DEPMIN,ALEN,
      2.   .           UEL, UWID, XSAREA, AREA, AWID, DFZ, VSET,
      3.   .     EL,ELEV,PELEV,RIVER,NEWQI,NEWTRB)
           .  .  IF (ECHO2) WRITE (6,9999)
 9999      .  .  FORMAT('*********** IN SERATRA TIME LOOP *********')
           .  .  IF(ECHO2) WRITE (6,4) ISEG, NXEQ
           .  .  IF (ECHO3)
           .  .  .  WRITE (6,1000)
 1000      .  .  .  FORMAT('0AFTER ICFLO, CCIN')
           .  .  .  WRITE (6,1010)
 1010      .  .  .  FORMAT('0',2X,'NODE NO.',2X,3(3X,'CONC. OF',4X),3(1X,'CONC.'
      1.   .  .  .    ' ASSOC.',2X),2X,'CONTAMINANT'/1X,'FROM BOTTOM',1X,
      2.   .  .  .  'SUSPENDED SAND',1X,'SUSPENDED SILT',1X,'SUSPENDED CLAY',
      3.   .  .  .  3X,'WITH SAND',6X,'WITH SILT',6X,'WITH CLAY',4X,'DISSOLVED',
      4.   .  .  .  ' CONC')
           .  .  .  WRITE(6,1020)(J,(CCIN(J,K),K=1,MAXCON),J=1,NELEM+1)
           .  .  .  IF (FERROR) REPORT-FATAL-ERROR-AND-STOP
           .  .  ...FIN
 1020      .  .    FORMAT(2X,I5,2X,1P7E15.5)
           .  .  IF (ECHO6)
           .  .  .  WRITE(6,1560)PTDPTH,NELMPT,PTDELZ,DEPTH,NELEM,DELZ
           .  .  .  WRITE(6,1570)
           .  .  .  WRITE(6,1010)
           .  .  .  WRITE(6,1020) (J,(C(J,K),K=1,MAXCON),J=1,NELMPT+1)
           .  .  .  WRITE(6,1580)
           .  .  .  WRITE(6,1110)
           .  .  .  WRITE(6,1020) (J,(COLD(J,K),K=1,MAXCON),J=1,NELMPT)
 1560      .  .  .  FORMAT('0IMMEDIATELY PRIOR TO RDSFLO, FOLLOWING ICFLO'/
      1.   .  .  .   'OPTDPTH =',E12.4,'  NELMPT =',I5,'  PTDELZ =',E12.4/
      2.   .  .  .   ' DEPTH  =',E12.4,'  NELEM  =',I5,'  DELZ   =',E12.4)
 1570      .  .  .  FORMAT('0NODAL CONCENTRATIONS PRIOR TO RDSFLO')
```

```
 1580 . . . FORMAT('0ELEMENT AVERAGE CONCENTRATIONS PRIOR TO RDSFLO')
      . . ...FIN
      . . IF (DEPTH .GT. DEPMIN)
      . . . T = ABS(PTDPTH/DEPTH-1.)
      . . . IF (NXEQ .GT. 1 .AND. T .GT. ZERO)
      . . . . CALL RDSFLU(ALEN,AREA,AWID,C,DELZ,DFZ,NELEM,NELMPT,
     1. . . . . PTDELZ,PWID,PXSAR,VSET,XSAREA)
      . . . . DO(J=1,MAXCON)
      . . . . . DO(K=1,NELEM)
      . . . . . . COLD(I,J)=(C(I,J)+C(I+1,J))/2.
      . . . . . ...FIN
      . . . . ...FIN
      . . . . IF (ECHO6)
      . . . . . WRITE(6,1590)
      . . . . . WRITE(6,1010)
      . . . . . WRITE(6,1020) (J,(C(J,K),K=1,MAXCON),J=1,NELEM+1)
      . . . . . WRITE(6,1600)
      . . . . . WRITE(6,1110)
      . . . . . WRITE(6,1020) (J,(COLD(J,K),K=1,MAXCON),J=1,NELEM)
 1590 . . . . . FORMAT('0NODAL CONCENTRATIONS FOLLOWING RDSFLO')
 1600 . . . . . FORMAT('0ELEMENT AVERAGE CONCENTRATIONS FOLLOWING RDSFLO')
      . . . . ...FIN
      . . . ...FIN
      . . . COMPUTE-BED-AND-WATER-SURFACE-ELEVATIONS
      . . ...FIN
C     . .
C     . . *** AVERAGE THE INFLOW CONCENTRATIONS INTO THE SEGMENT BY TAKING
C     . .        INTO ACCOUNT THE TRIBUTARY INPUT.
C     . .
      . . IF (DEPTH .GT. DEPMIN)
      . . . WHEN (NTRIBS .GT. 0)
      . . . . IF (NEWTRB .OR. NEWIC .OR. NEWQI)
      . . . . . DO (K=1,MAXCON)
      . . . . . . DO (J=1,NELEM+1)
      . . . . . . . CDUMMY(J) = CCIN(J,K)
      . . . . . . ...FIN
      . . . . . . DO (J=1,NELEM)
      . . . . . . . CMASS=(CDUMMY(J)+CDUMMY(J+1))/2.*QHIN(J)
      . . . . . . . CMASS=(CMASS+CTRB(J,K)*SECDAY)/QHIN(J)
C     . . . . . . .
C     . . . . . . .   NOTE: CMASS IS IN (KG/M**3)
C     . . . . . . .
      . . . . . . . WHEN (J.EQ.1)
      . . . . . . . . WHEN (K.EQ.7)
      . . . . . . . . . CCIN(1,K)=CMASS
      . . . . . . . . . CCIN(2,K)=CMASS
      . . . . . . . . ...FIN
      . . . . . . . . ELSE
      . . . . . . . . . KK=K
      . . . . . . . . . IF(K.GT.3) KK=KK-3
      . . . . . . . . . COEF=0.
      . . . . . . . . . WS=VSET(KK)*AREA(1)/(AWID(1)*ALEN)
      . . . . . . . . . EZ=DFZ(KK)
      . . . . . . . . . CCIN(1,K)=(2.*CMASS-COEF*DELZ/EZ)/(2.-WS*DELZ/EZ)
      . . . . . . . . . CCIN(2,K)=2.*CMASS - CCIN(1,K)
      . . . . . . . . ...FIN
```

```
                .  .  .  .  .  .  .  ...FIN
                .  .  .  .  .  .  .  ELSE
                .  .  .  .  .  .  .  .  CCIN(J+1,K)=2.*CMASS - CCIN(J,K)
                .  .  .  .  .  .  .  ...FIN
                .  .  .  .  .  .  ...FIN
                .  .  .  .  .  ...FIN
                .  .  .  .  ...FIN
                .  .  .  .  IF (ECHO3)
                .  .  .  .  .  WRITE(6,1300)
         1300   .  .  .  .  .  FORMAT('0AFTER TRIBUTARY INFLOW, CCIN AND QHIN')
                .  .  .  .  .  WRITE(6,1310)
         1310   .  .  .  .  .  FORMAT('0ELEMENT/NODE',3(3X,'CONC. OF',4X),3(1X,'CONC.'
               1.  .  .  .  .  ' ASSOC.',2X),2X,'CONTAMINANT',7X,'FLOW'/1X,'FROM BOTTOM',
               2.  .  .  .  .  2X,'SUSPENDED SAND SUSPENDED SILT SUSPENDED CLAY   WITH ',
               3.  .  .  .  .  'SAND',6X,'WITH SILT',6X,'WITH CLAY    DISSOLVED CONC.',
               4.  .  .  .  .  2X,'(M**3/SEC)')
                .  .  .  .  .  WRITE(6,1320)(J,(CCIN(J,K),K=1,MAXCON),QHIN(J),J=1,NELEM+1)
                .  .  .  .  ...FIN
         1320   .  .  .  .  FORMAT(2X,I5,4X,1P8E15.5)
                .  .  .  ...FIN
                .  .  .  ELSE
                .  .  .  .  DO (J=1,MXELEM)
                .  .  .  .  .  DO (K=1,MAXCON) CTRB(J,K)=0.0
                .  .  .  .  ...FIN
                .  .  .  .  QTR=0.0
                .  .  .  ...FIN
         C      .  .  .
         C      .  .  .  *** DETERMINE THE PROPER PHOTOLYSIS COEFFICIENT TO BE USED ***
                .  .  .  JULSEC = JULSEC + DELTH
                .  .  .  NSP = MOD(JULSEC,SECYR)
                .  .  .  CONDITIONAL
                .  .  .  .  (NSP .GE.        1 .AND. NSP .LE.  7948800) IPC = 1
                .  .  .  .  (NSP .GE.  7948801 .AND. NSP .LE. 15897600) IPC = 2
                .  .  .  .  (NSP .GE. 15897600 .AND. NSP .LE. 23760000) IPC = 3
                .  .  .  .  (NSP .GE. 23760001 .AND. NSP .LE. SECYR) IPC = 4
                .  .  .  .  (NSP .EQ. 0) IPC = 4
                .  .  .  ...FIN
                .  .  ...FIN
                .  .  WHEN (DEPTH .LE. DEPMIN)
                .  .  .  DO (L=1,MAXCON)
                .  .  .  .  DO (I=1,MXELEM)
                .  .  .  .  .  C(I,L)=0.
                .  .  .  .  .  COLD(I,L)=0.
                .  .  .  .  ...FIN
                .  .  .  ...FIN
                .  .  .  IF (DECAY(1) .GT. 0.0)
                .  .  .  .  CALL BEDDK(B, DECAY,DELTD,NBED)
                .  .  .  ...FIN
                .  .  ...FIN
                .  .  ELSE
                .  .  .  FERROR = .FALSE.
         C      .  .  .
                .  .  .  PCOEFF=PCOEF(IPC)
                .  .  .  IF (ECHO5)
                .  .  .  .  WRITE(6,1700)
                .  .  .  .  WRITE(6,1710)
```

```
                 .  .  .  .  WRITE(6,1720) (I,AREA(I),ABAR(I),AWID(I),BWID(I),QV(I),QHIN(I),
              1. .  .  .  .                      QHOUT(I),I=1,NELEM+1)
         1700 .  .  .  .  .  FORMAT('0GEOMETRY AND DISCHARGE INFO PRIOR TO TRANSP')
         1710 .  .  .  .  .  FORMAT('0ELEMENT/NODE',6X,'AREA',9X,'ABAR',11X,'AWID',11X,'BWID',
              1. .  .  .  .          12X,'QV',12X,'QHIN',11X,'QHOUT')
         1720 .  .  .  .  .  FORMAT(2X,I5,3X,1P7E15.4)
                 .  .  .  .  ...FIN
                 .  .  .  CALL TRANSP(FERROR,PCOEFF,BWID,AWID,ABAR,DEPTH,OLDC,ECHO4,
              1. .  .  .                 CROSEC,ECHO7, ECHO8, ECHO9)
       C*****************************************************************
       C         .  .  .  CHANGE 4/2/81
       C         .  .  .
                 .  .  .  IF(ECHO5)WRITE(6,2000)((C(I,J),J=1,MAXCON),I=1,NELEM+1)
         2000    .  .  .  FORMAT('0AFTER TRANSP'/50(1X,1P7E15.5/))
       C         .  .  .
       C**********************************************************************
       C         .  .  .
                 .  .  .  IF (FERROR) REPORT-FATAL-ERROR-AND-STOP
       C         .  .  .
                 .  .  .  COMPUTE-BED-AND-WATER-SURFACE-ELEVATIONS
       C         .  .  .
                 .  .  ...FIN
       C         .  .
       C         .  .  *** SAVE THE RESULTS OF THIS TIME STEP, IT WILL BECOME INPUT TO
       C         .  .     THE NEXT SEGMENT ***
                 .  .  WRITE(OUTFLO) DEPTH,DELZ,NELEM,(QHOUT(K),XSAREA(K),AWID(K),
              1. .                 K=1,NELEM),((C(L,K),K=1,MAXCON),L=1,NELEM+1),
              2. .                ((OLDC(L,K),K=1,MAXCON),L=1,NELEM+1)
       C         .  .
       C*****************************************************************
       C         .  .  CHANGE 7/27/81
       C         .  .
                 .  .  WRITE(9) DEPTH,DELZ,NELEM,(QHOUT(K),XSAREA(K),AWID(K),
              1. .                 K=1,NELEM),((C(L,K),K=1,MAXCON),L=1,NELEM+1),
              2. .                ((OLDC(L,K),K=1,MAXCON),L=1,NELEM+1)
       C         .  .
       C*****************************************************************
       C         .  .
       C         .  .
                 .  .  IF (ANALYS) SAVE-THE-RESULTS-FOR-TIME-SERIES-ANALYSIS
       C         .  .
                 .  .  IF (MOD(NXEQ,IIPRT) .EQ. 0)
       C         .  .  .  *** SAVE THE RESULTS FOR PRINTING AND OTHER POST PROCESSING ***
                 .  .  .  CALL SAVEIT(H, BDIV, BED, ELEV, C, DELZ,  NBED, NELEM,
              1. .  .  .                NXEQ, RESELN, STRESS, XYSO, OLDC,
              2. .  .  .    ALEN,QHIN,QHOUT,CCIN,QV,AWID,BWID,VSET,DENS,DELTD,DFZ,
              3. .  .  .           POR,TBED)
                 .  .  ...FIN
       C         .  .
                 .  .  NELMPT=NELEM
                 .  .  PTDPTH=DEPTH
                 .  .  PTDELZ=DELZ
                 .  .  DO (I=1,NELEM)
                 .  .  .  PXSAR(I)=XSAREA(I)
                 .  .  .  PWID(I)=AWID(I)
                 .  .  ...FIN
```

```
      C     .  .
      C     .  .  *** CHECK SENSE SWITCH #2 TO SEE IF THE RUN IS TO BE STOPPED ***
            .  .  IF (IS2.EQ.1)
            .  .  .  CLOSE-THE-OPEN-FILES
            .  .  .  OPEN(UNIT=1,NAME='TT:')
            .  .  .  WRITE(1,2) NXEQ,ISEG
       2    .  .  .  FORMAT(//21X,'***** SERATRA *****'/
          1.  .  .    5X,'TERMINATED BY OPERATER AFTER TIME PLANE #',I10/
          2.  .  .    5X,'IN SEGMENT #',I5)
            .  .  .  STOP
            .  .  ...FIN
      C     .  .                    END OF TIME STEP LOOP
            .  .  NXEQ = NXEQ + 1
            .  ...FIN
      C     .                      END OF SEGMENT LOOP
            .  PELEV = ELEV
            .  NFRST = 1
            .  LUNTMP=INFLO
            .  INFLO=OUTFLO
            .  OUTFLO=LUNTMP
            .  REWIND INFLO
            .  REWIND OUTFLO
            .  CLOSE(UNIT=5)
            .  CLOSE(UNIT=9)
            ...FIN
            CLOSE-THE-OPEN-FILES
            STOP
      C

      ----------------------------------------

            TO COMPUTE-BED-AND-WATER-SURFACE-ELEVATIONS
            .  BELEV = ELEV + BED
            .  RESELN = DEPTH + BELEV
            ...FIN

      ----------------------------------------

            TO CLOSE-THE-OPEN-FILES
            .  CLOSE(UNIT=1)
            .  CLOSE(UNIT=2)
            .  CLOSE(UNIT=3)
            .  CLOSE(UNIT=4)
            .  CLOSE(UNIT=5)
            .  CLOSE(UNIT=9)
            .  CLOSE(UNIT=6)
            .  CLOSE(UNIT=7)
            .  CLOSE(UNIT=8)
            ...FIN

      ----------------------------------------

            TO REPORT-FATAL-ERROR-AND-STOP
            .  CLOSE-THE-OPEN-FILES
            .  OPEN (UNIT=1,NAME='TT:')
            .  WRITE(1,1)
```

```
 1       .  FORMAT(//10X,'**** SEMATRA -- FATAL ERROR *****'/
        1.    ' PRINT "SED.LST" FOR DETAILS')
 C       .  *** THE IF STATEMENT BELOW IS A CONCESSION TO THE COMPILER ***
         .  IF(FERROR) STOP
         ...FIN

 --------------------------------------

         TO RESET-DATA-TIME-CONTROLS
         .  ENDIC = 0
         .  ENDHYD = 0
         .  ENDTRB = 0
         ...FIN

 --------------------------------------

         TO SAVE-THE-RESULTS-FOR-TIME-SERIES-ANALYSIS
 C       .
 C       .  *** COMPUTE THE VOLUME OF EACH ELEMENT AND THE TOTAL VOLUME
 C       .       OF THE WATER COLUMN ***
 C       .
         .  AVOL=0.
         .  DO (I=1,NELEM)
         .  .  ELMVOL(I) = DELZ*ABAR(I)
         .  .  AVOL=AVOL+ELMVOL(I)
         .  ...FIN
 C       .
 C       .  *** AVERAGE DISSOLVED (KG/M**3) ***
         .  AVGDIS = 0.0
         .  DO(I=1,NELEM)AVGDIS=AVGDIS+(C(I,7)+C(I+1,7))*ELMVOL(I)/2.
         .  AVGDIS = AVGDIS / AVOL
         .  IF (AVGDIS .GT. ANALMT)
 C       .  .
 C       .  .  *** AVERAGE SEDIMENT (KG/M**3) ***
         .  .  AVGSED = 0.0
         .  .  DO(I=1,NELEM)
         .  .  .  AVGSED=AVGSED+(C(I,1)+C(I+1,1)+C(I,2)+C(I+1,2)+
        1.  .  .    C(I,3)+C(I+1,3))/2.
         .  .  ...FIN
         .  .  AVGSED = AVGSED / AVOL
 C       .  .
 C       .  .  **** AVERAGE (PARTICULATE (PC/KG)*SEDIMENT(KG/M**3))
         .  .  PARPCM = 0.0
         .  .  DO (I=1,NELEM)
         .  .  .  PARPCM = PARPCM +ELMVOL(I)*(C(I,4)+C(I,5)+C(I,6)
        1.  .  .    +C(I+1,4)+C(I+1,5)+C(I+1,6))/2.
         .  .  ...FIN
         .  .  PARPCM = PARPCM / AVOL
 C       .  .
 C       .  .  *** AVERAGE PARTICULATE (PC/KG) ***
         .  .  PARPCK = PARPCM / AVGSED
 C       .  .
         .  .  TOTKG = ( PARPCM + AVGDIS )*AVOL
 C       .  .
         .  .  CAVGMX = MAX(CAVGMX, AVGDIS)
         .  .  IFLOW = 0.0
```

```
           .  .  DO (I=1,NELEM) TFLOW = TFLOW + PIQAVG(I)
           .  .  TFLOW = TFLOW / SECDAY
           .  .  WRITE(8) ISEG,NXEQ,TFLOW,AVGSED,AVGDIS,PARPCM,PARPCK,TOTKG
           .  ...FIN
           ...FIN
      C
           END
```

PROCEDURE CROSS-REFERENCE TABLE

```
00554 RESET-DATA-TIME-CONTROLS
      00158

00559 SAVE-THE-RESULTS-FOR-TIME-SERIES-ANALYSIS
      00485

00534 CLOSE-THE-OPEN-FILES
      00505  00527  00546

00530 COMPUTE-BED-AND-WATER-SURFACE-ELEVATIONS
      00357  00465

00545 REPORT-FATAL-ERROR-AND-STOP
      00142  00186  00284  00299  00319  00463
```

(FLECS VERSION 22.46)

```
        ------------------------------------
              SUBROUTINE SETUP(II, DD, P, S, R, VEL, NELEM,ECHO4,WIDTH)
        C
        C     THIS SUBROUTINE SETS UP THE FINITE ELEMENT MATRICES
        C
        C     CALLED BY TRANSP.
        C
              LOGICAL*1 ECHO4
              INCLUDE 'SY:ELMSIZ.PRM'
        C
              DIMENSION DD(10), P(MXELEM,3), PEL(2,2), R(MXELEM), S(MXELEM,3),
             1          SEL(2,2)
        C
        C
              PEL(1,1) = 1./3.
              PEL(1,2) = 1./6.
              PEL(2,1) = 1./6.
              PEL(2,2) = 1./3.
              SEL(1,1)= DD(1) + DD(3) + DD(7)/3.
              SEL(1,2)= -DD(1) + DD(4) + DD(7)/6.
              SEL(2,1)= -DD(1) - DD(5) + DD(7)/6.
              SEL(2,2)= DD(1) + DD(6) + DD(7)/3.
              IF(II.EQ.1) SEL(1,1)=SEL(1,1)+VEL
              IF(II.EQ.NELEM) SEL(2,2)=SEL(2,2)+VEL
        C
              DO (I=1,2)
              .  DO (J=1,2)
              .  .  PEL(I,J) = PEL(I,J) * WIDTH
              .  .  SEL(I,J) = SEL(I,J) * WIDTH
              .  ...FIN
              ...FIN
              DO (I=8,10,1) DD(I) = DD(I) * WIDTH
              DD2=DD(8)/2.0
              DO (J=1,2)
              .  NR=II + J - 1
              .  DO (K=1,2)
              .  .  MC=2*(II + K - 1) - NR
              .  .  P(NR,MC)= P(NR,MC) + PEL(J,K)
              .  .  S(NR,MC)= S(NR,MC) + SEL(J,K)
              .  ...FIN
              .  R(NR)= R(NR) + DD2 + DD(8+J)
              ...FIN
              IF (ECHO4)
              .  WRITE(6,999)
          999 .  FORMAT(' *************************IN SETUP')
              .  WRITE(6,1000)(I,(SEL(I,J),J=1,2),I=1,2)
              .  WRITE(6,1100)
              .  WRITE(6,1200)(I,(P(I,J),J=1,3),(S(I,J),J=1,3),R(I),I=1,NELEM+1)
         1000 .  FORMAT(' SEL(I,J), I=',I2,1X,' J=1,2 ',5X,1P2E14.4)
         1100 .  FORMAT('0NODE',7X,'P(I,1)',10X,'P(I,2)',10X,'P(I,3)',10X,
             1.  'S(I,1)',10X,'S(I,2)',10X,'S(I,3)',10X,'R(I)')
         1200 .  FORMAT(1X,I3,1P7E16.5)
              ...FIN
              RETURN
```

```
 00054         END
         (FLECS VERSION 22.46)
         ----------------------------------------
```

```
      ---------------------------------------

            SUBROUTINE SHEARR(DEPTH, D50, STRESS, USTAR, VEL)
      C
      C   THIS SUBROUTINE CALCULATES BED SHEAR STRESS AND SHEAR VELOCITY FOR
      C   A SEDIMENT LADEN FLOW. METHOD IS APPLICABLE FOR RESERVOIRS.
      C   REF. HYDRAULICS OF SEDIMENT TRANSPORT BY W.H. GRAF, EQ 8.49
      C
      C   FORMAL PARAMETERS:
      C     DEPTH  - FLOW DEPTH (METERS)
      C     D50    - MEDIAN BED SEDIMENT DIAMETER (METERS)
      C     STRESS - BED SHEAR STRESS (KG/M**2)
      C     USTAR  - SHEAR VELOCITY (M/SEC)
      C     VEL    - AVERAGE VELOCITY (M/SEC)
      C
      C   CALLED BY: HYDFLO, ICFLO
      C
      C   RHO - WATER DENSITY (KG(FORCE)/M**3)
            DATA RHO /1000,/
      C
      C   AKAPPA - KARMAN CONSTANT
            DATA AKAPPA /0,4/
      C
            USTAR=VEL/(17,66+(ALOG10(DEPTH/(96,5*D50)))*2,3/AKAPPA)
            STRESS=RHO*USTAR**2,0/9,8
            RETURN
            END

      (FLECS VERSION 22,46)

      ---------------------------------------
```

```
(FLECS VERSION 22.46)      19-MAR-81   14:27:15  PAGE  00001

        ---------------------------------------

              SUBROUTINE SHEARS(ALEN,ELEV, HRAD, PELEV,SLOPE,STRESS,USTAR)
        C
        C    THIS METHOD OF COMPUTING BED SHEAR STRESS AND SHEAR VELOCITY
        C    IS APPLICABLE TO RIVERS AND STREAMS.
        C
        C    FORMAL PARAMETERS:
        C      ALEN    - SEGMENT LENGTH
        C      ELEV    - ELEVATION OF THE CURRENT SEGMENT
        C      HRAD    - HYDRAULIC RADIUS OF THE SEGMENT
        C      PELEV   - ELEVATION OF THE PREVIOUS SEGMENT
        C      SLOPE   - BED SLOPE
        C      STRESS  - BED SHEAR STRESS
        C      USTAR   - SHEAR VELOCITY
        C
        C    CALLED BY:HYDFLO
        C
        C      G    - GRAVITY (M/S**2)
        C      RHO - DENSITY OF WATER (KG(FORCE)/M**3)
              DATA RHO/1000./
              DATA   G/9.801/
              SLOPE = (PELEV - ELEV) / ALEN
              STRESS = SLOPE * RHO * HRAD
              USTAR = SQRT(G * SLOPE * HRAD)
        C
              RETURN
              END

        (FLECS VERSION 22.46)

        ---------------------------------------
```

```
        -------------------------------------
              SUBROUTINE SILCLA(ABAR, B, BDIV,  CCIN, DELTD, DELZ, DEPTH,
             1                  DENS, DSHR, ERODE, HRAD, J1, NBED, COLD,
             2                  NELEM, POR, QHIN, QHOUT, SCSHR, STRESS, VSET,
             3                  XYSO, DEPO, ILAYR, SD, SR, XNT,
             4                  CROSEC, BWID, ALEN, ECHO7, SCOUR)
        C
        C     THIS SUBROUTINE COMPUTES THE RATE AND SOURCE TERMS FOR THE
        C     TRANSPORT OF SILT (J1=2) AND CLAY (J1=3)
        C
        C     INPUT PARAMETERS:
        C        ABAR   - AVERAGE AREA
        C        B      - BED CONDITIONS
        C        BDIV   - STANDARD BED LAYER THICKNESS
        C        C      - WATER CONDITIONS
        C        CCIN   - CONCENTRATION OF INFLOW
        C        DELTD  - TIME STEP (DAYS)
        C        DELZ   - STANDARD ELEMENT THICKNESS
        C        DENS   - DENSITY
        C     DEPTH  - DEPTH OF FLOW
        C        DSHR   - CRITICAL SHEAR STRESS FOR DEPOSITION
        C        ERODE  - ERODABILITY, (KG/M**2/SEC)
        C        HRAD   - HYDRAULIC RADIUS
        C        J1     - =1; SILT      =2; CLAY
        C        NBED   - NUMBER OF BED LAYERS
        C        NELEM  - NUMBER OF ELEMENTS
        C        POR    - POROSITY
        C        QHIN   - INFLOW DISCHARGE
        C        QHOUT  - OUTFLOW DISCHARGE
        C        SCSHR  - CRITICAL SHEAR STRESS FOR SCOUR
        C        STRESS - BED SHEAR STRESS
        C        VSET   - PARTICLE SETTLING VELOCITY
        C        XYSO   - THICKNESS OF TOP BED LAYER
        C     OUTPUT PARAMETERS:
        C        ILAYR  - NO. OF BED LAYERS AFFECTED BY DEPOSITION AND EROSION
        C        SD     - DEPOSITION RATE, (KG(PC)/M**3/DAY)
        C        SR     - EROSION RATE, (KG(PC )/M**3/DAY)
        C        XNT    - WEIGHT OF TOP BED SEDIMENT LAYER, (KG/M**2)
        C     DEPO   - BED DEPOSITION RATE (KG(PC)/M2/DAY)
        C     SCOUR  - BED SCOUR RATE (KG(PC)/M2/DAY)
        C
        C     CALLED BY: TRANSP
        C     CALLS: DEPCAL
        C
              INCLUDE 'SY:ELMSIZ.PRM'
        C
              REAL K4FUNC,K4
              LOGICAL*1 ECHO7
        C
              DIMENSION ABAR(MXELEM), B(MAXLEV,MAXCON-1),COLD(MXELEM,MAXCON),
             1          DENS(3), DSHR(3), ERODE(3), ILAYR(3),
             2          QHIN(MXELEM), QHOUT(MXELEM), SCSHR(3),
             3          SD(MXELEM,6), SR(6), VSET(3), XNT(3),
             4          CCIN(MXELEM,MAXCON), BWID(MXELEM), DEPO(6), SCOUR(6)
```

```
                DATA SECDAY/86400./
      C
                J2=J1+3
                XNT(J1)=0.
                IF (NBED.GT.0)
                .  XDTOP=(1.0-POR)/(B(NBED,1)/DENS(1)+B(NBED,2)/DENS(2) +
               1.          B(NBED,3)/DENS(3))
                .  XNT(J1) = XDTOP * B(NBED,J1) * XYSQ
                ...FIN
                DEPO(J1) = 0.0
                DEPO(J2) = 0.0
                SR(J1) = 0.0
                SR(J2) = 0.0
                SCOUR(J1) = 0.0
                SCOUR(J2) = 0.0
                RS = 0.0
                CS = 0.0
                VOLUME = CROSEC * ALEN
                DO (IX = 1,NELEM)
                .  SD(IX,J1) = 0.0
                .  SD(IX,J2) = 0.0
                ...FIN
                ILAYR(J1) = 0
                IF(ECHO7)
                .  IF (STRESS .LT. DSHR(J1))
      C         .  .
      C     SEDIMENT DEPOSITION
      C         .  .
                .  .  ILAYR(J1) = -1
                .  .  AVGC = 0.0
                .  .  TOTQ = 0.0
                .  .  DO (IX = 1,NELEM)
      C         .  .  .
      C         .  .  .  IT IS IMPLICITLY ASSUMED THAT A DOWNSTREAM COURANT NUMBER
      C         .  .  .  AT OR NEAR UNITY IS EMPLOYED IN THIS ANALYSIS
      C         .  .  .
                .  .  .  TOTQ = TOTQ + (QHIN(IX)+QHOUT(IX))/2.
                .  .  .  AVGC=AVGC+QHIN(IX)*(CCIN(IX,J1)+CCIN(IX+1,J1))/4.
               1.  .  .          +QHOUT(IX)*COLD(IX,J1)/2.
                .  .  ...FIN
                .  .  AVGC = AVGC / TOTQ
                .  .  DEPO(J1) = VSET(J1) * AVGC * (1.0-(STRESS/DSHR(J1)))
                .  .  RATE = DEPO(J1) * BWID(1) * ALEN / VOLUME
                .  .  DO (K = 1,NELEM)
                .  .  .  SD(K,J1) = RATE
                .  .  .  VOLK = ABAR(K) * DELZ
                .  .  .  SED=QHIN(K)*(CCIN(K,J1)+CCIN(K+1,J1))/4.+QHOUT(K)*COLD(K,J1)/2.
                .  .  .  CONT=QHIN(K)*(CCIN(K,J2)+CCIN(K+1,J2))/4.+QHOUT(K)*COLD(K,J2)/2.
                .  .  .  RATEK = RATE * VOLK
                .  .  .  SD(K,J2) = RATEK * CONT / SED / VOLK
                .  .  .  DEPO(J2) = DEPO(J2) + SD(K,J2) * VOLK / BWID(1)
                .  .  ...FIN
                .  ...FIN
                .  IF (STRESS .GT. SCSHR(J1).AND. NBED. GT . 0)
      C     SEDIMENT SCOURING
      C         .  .
```

```
(FLECS VERSION 22.46)      13-MAR-81    13:29:56  PAGE  00003
            .  .  RS = ERODE(J1) * SECDAY * (STRESS / SCSHR(J1) - 1.0)
            .  .  ILAYR(J1) = 0
     C      .  .
     C      .  .  COMPUTE AVAILABILITY OF COHESIVE SEDIMENT IN BED LAYERS.
     C      .  .  MAXIMUM NUMBER OF LAYERS SCOURED IS RESTRICTED BY SAND SCOURING.
     C      .  .
            .  .  RS = RS * DELTD
            .  .  WHEN (.NOT.(RS .GT. XNT(J1) .AND. ILAYR(1) .GT. 0) )
            .  .  .  RS = AMIN1(RS,XNT(J1))
            .  .  .  CS = RS * B(NBED,J2)
            .  .  .  XNT(J1) = XNT(J1)-RS
            .  .  ...FIN
            .  .  ELSE
            .  .  .  FAC=B(NBED,J2)
            .  .  .  RSUSP = RS
            .  .  .  RS = 0.
     140    .  .  .  ILAYR(J1) = ILAYR(J1)+1
            .  .  .  NB = NBED-ILAYR(J1)
            .  .  .  RSUSP = RSUSP-XNT(J1)
            .  .  .  RS = RS+XNT(J1)
            .  .  .  CS = CS + FAC*XNT(J1)
            .  .  .  XNT(J1) = 0.0
            .  .  .  FAC = 0.0
            .  .  .  IF (NB.NE.0)
            .  .  .  .  XND=(1.0-POR)/(B(NB,1)/DENS(1)+B(NB,2)/DENS(2)+B(NB,3)/DENS(3))
            .  .  .  .  XNT(J1) = BDIV * B(NB,J1) * XND
            .  .  .  .  FAC = B(NB,J2)
            .  .  .  ...FIN
     C      .  .  .
            .  .  .  IF (ILAYR(J1).EQ.ILAYR(1)) GO TO 155
            .  .  .  IF (RSUSP.GE.XNT(J1).AND.ILAYR(1).GT.ILAYR(J1))
            .  .  .  .  GO TO 140
            .  .  .  ...FIN
     C      .  .  .
     155    .  .  .  CONTINUE
            .  .  .  DEL = AMIN1(RSUSP,XNT(J1))
            .  .  .  RS = RS + DEL
            .  .  .  CS = CS + DEL * FAC
            .  .  .  XNT(J1) = XNT(J1) - DEL
            .  .  ...FIN
            .  ...FIN
            .  SCOUR(J1) = RS / DELTD
            .  SCOUR(J2) = CS / DELTD
            .  SR(J1) = RS / DELTD * BWID(1) * ALEN / VOLUME
            .  SR(J2) = CS / DELTD * BWID(1) * ALEN / VOLUME
     C      .
            ...FIN
            RETURN
            END

     (FLECS VERSION 22.46)

     ----------------------------------------
```

```
        ------------------------------------
            SUBROUTINE STRTUP(BASE, DEV, ECHO, FNAME, FTYPE, GUIC, INFLO,
           1                     ISTRT, NFRST, OUTFLO, SMETH, UUIC,
           2     SAVECH, JSEG)
      C
      C   THIS ROUTINE IS RESPONSIBLE FOR THE INTERACTIVE I/O AND OPENING THE
      C   PROPER FILES.
      C
      C
      C   FORMAL PARAMETERS:
      C      BASE    - 5-CHARACTER BASE FILE NAME (BYTE)
      C      DEV     - BASE OUTPUT FILE DEVICE (BYTE)
      C      ECHO    -  LINE PRINTER ECHO CONTROL VARIABLE (L*1)
      C      SAVECH(1)- ECHO CONTROL FOR SERATRA HEADINGS
      C      SAVECH(2)- ECHO CONTROL FOR INFLUENT CONCENTRATIONS
      C      SAVECH(3)-  ECHO CONTROL FOR SETUP OF ELEMENT MATRIX
      C     SAVECH(4)-  ECHO CONTROL FOR GEOMETRY AND CONCENTRATIONS
      C     SAVECH(5)-  ECHO CONTROL FOR ROSFLO, BEFORE AND AFTER
      C     SAVECH(6)-  ECHO CONTROL FOR SEDIMENTATION
      C     SAVECH(7)  -  ECHO CONTROL FOR DIAGNOSTICS WITHIN  SAND, SILCLA
      C     SAVECH(8)-  ECHO CONTROL FOR PRINTOUT OF SCOUR/DEPOSITION DETAIL
      C     SAVECH(9)-  ECHO CONTROL - UNDEFINED - TRANSFERS TO TRANSP
      C      SAVECH(10)- ECHO CONTROL - UNDEFINED
      C      FNAME   -  FILE DESCRIPTION FOR THE RESULT FILE (BYTE)
      C      FTYPE   - BASE OUTPUT FILE EXTENSION (BYTE)
      C      GUIC    - GROUP NUMBER FROM UIC OF BASE FILE NAME (BYTE)
      C      INFLO   - LUN NUMBER TO THE DATA FROM THE PREVIOUS SEGMENT
      C      ISTRT   - STARTING SEGMENT NUMBER
      C      NFRST   - STARTING TIME PLANE NUMBER (I*4)
      C      OUTFLO  - LUN NUMBER TO THE FILE RECEIVING THE RESULTS OF THE
      C                CURRENT SEGMENT (I*2)
      C      SMETH   - METHOD TO BE USED TO CALCULATE THE SAND CAPACITY (BYTE)
      C      UUIC    - USER NUMBER FROM UIC OF BASE FILE NAME (BYTE)
      C
      C   CALLED BY: SERATRA
      C   CALLS: FDCODE
      C
      C
            BYTE ANSWER,YES,R,FNAME(29),FTYPE(3),DEV(3),GUIC(3),UUIC(3),
           1     BASE(5),SMETH,INPFIL(30)
      C
            INTEGER*2 OUTFLO
      C
            INTEGER*4 NFRST
      C
            LOGICAL*1 ECHO, SAVECH, WRTSEG
      C
            DIMENSION JSEG(5), SAVECH(10)
      C
            DATA R/'R'/
            DATA YES/'Y'/
            DATA WRTSEG/.FALSE./
      C
            WRITE(R,1)
```

```
      READ(8,2) NCHR,(INPFIL(I),I=1,NCHR)
      INPFIL(NCHR+1) = 0
      OPEN(UNIT=1,NAME=INPFIL,TYPE='OLD',READONLY)
C
      WRITE(8,3)
      READ(8,4)ANSWER
      IF (ANSWER .EQ. YES) ECHO = .TRUE.
C
      WRITE(8,6)
      READ(8,4)(FNAME(I),I=1,29)
      CALL FDCODE(FNAME,BASE,JSEG,FTYPE,DEV,GUIC,UUIC)
C
      WRITE(8,7)
      READ(8,4) SMETH
C
      WRITE(3,8)
      READ(8,4) ANSWER
      IF (ANSWER.EQ.YES) SAVECH(1) = .TRUE.
C
      WRITE(8,9)
      READ(8,4) ANSWER
      IF (ANSWER.EQ.YES) SAVECH(2) = .TRUE.
C
      WRITE(8,10)
      READ(8,4) ANSWER
      IF (ANSWER.EQ.YES) SAVECH(3) = .TRUE.
C
      WRITE(8,11)
      READ(8,4) ANSWER
      IF (ANSWER.EQ.YES) SAVECH(4) = .TRUE.
C
      WRITE (8,12)
      READ(8,4) ANSWER
      IF (ANSWER.EQ.YES) SAVECH(5) = .TRUE.
C
      WRITE (6,13)
      READ(8,4) ANSWER
      IF (ANSWER.EQ.YES) SAVECH(6) = .TRUE.
C
      WRITE (8,14)
      READ(8,4) ANSWER
      IF (ANSWER.EQ.YES) SAVECH(7) = .TRUE.
C
      WRITE (6,15)
      READ(8,4) ANSWER
      IF (ANSWER.EQ.YES) WRTSEG = .TRUE.
      WHEN (WRTSEG) JSEG(1) = 0
      ELSE
      .  WRITE (8,16)
      .  READ(8,17)  (JSEG(J),J=1,5)
      ...FIN
      CLOSE(UNIT=8)
      NFRST = 1
      ISTRT=1
      INFLO=2
      OUTFLO=3
```

```
         OPEN(UNIT=2,NAME='DUMMY.DT1',TYPE='NEW',FORM='UNFORMATTED')
         OPEN(UNIT=3,NAME='DUMMY.DT2',TYPE='NEW',FORM='UNFORMATTED')
         OPEN(UNIT=4,NAME='HYDROLOGY.TMP',TYPE='NEW',FORM='UNFORMATTED')
         OPEN(UNIT=7,NAME='TRIBUTARY.TMP',TYPE='NEW',FORM='UNFORMATTED')
         OPEN(UNIT=6,NAME='SED.LST',TYPE='NEW')
         RETURN
C
C        ** FORMATS **
      1  FORMAT('$ENTER NAME OF INPUT FILE>')
      2  FORMAT(Q,30A1)
      3  FORMAT('$DO YOU WANT THE INPUT FILE ECHOED (Y OR N)>')
      4  FORMAT(29A1)
      6  FORMAT('$ENTER BASE FILE NAME>')
      7  FORMAT(' WHICH SAND CAPACITY METHOD IS TO BE USED?'/
        1 '$ENTER T (TOFFALETTI) OR C (COLBY)>')
      8    FORMAT('$DO YOU WANT SERATRA HEADINGS ECHOED (Y OR N)?')
      9    FORMAT('$DO YOU WANT INFLUENT CONCENTRATIONS ECHOED (Y OR N)?')
     10    FORMAT('$DO YOU WANT ELEMENT MATRICES ECHOED (Y OR N)?')
   11    FORMAT('$DO YOU WANT GEOMETRY AND CONCENTRATIONS ECHOED (Y OR N)',
        1       '?')
   12    FORMAT('$DO YOU WANT CONCENTRATION ECHOED BEFORE AND AFTER ROSFLO'
        1        ,' (Y OR N)?')
   13    FORMAT('$DO YOU WANT SCOUR/DEPOSITION TO OCCUR? (Y OR N)')
   14    FORMAT('$DO YOU WANT COMPLETE SCOUR/DEPOSITION INFORMATION',
        1' RECORDED? (Y OR N)')
   15    FORMAT('$DO YOU WANT COMPLETE ECHO# INFORMATION FOR ALL',
        1' SEGMENTS? (Y OR N)')
   16    FORMAT('$FOR WHICH SEGMENTS DO YOU WANT COMPLETE ECHO#',
        1' INFORMATION? (MAXIMUM OF 5)')
   17    FORMAT(5I5)
C
         END
```

(FLECS VERSION 22.46)

```
        ------------------------------------
      SUBROUTINE TOFFAL(ALEN, D50, GSI, HRAD, QTOT, SLOPE, TEMPR, VOL,
     1                  VSET, GSU, GSM, GSL, GSB, YU, YM, YL)
C
C   THIS SUBROUTINE USES TOFFALETTI'S METHOD TO CALCULATE THE CAPACITY
C   OF THE FLOW TO TRANSPORT SAND.  A SUMMARY OF THIS METHOD CAN BE
C     FOUND IN THE ASCE 1975 EDITION OF "SEDIMENTATION ENGINEERING"
C     PAGES 209 - 213.
C
C     FORMAL PARAMETERS:
C       ALEN  - SEGMENT LENGTH
C       D50   - MEDIAN BED SEDIMENT DIAMETER (METERS)
C       GSI   - TOTAL CAPACITY OF THE SEGMENT (KG/DAY/M)
C       HRAD  - HYDRAULIC RADIUS
C       QTOT  - TOTAL FLOW WITHIN THE SEGMENT
C       SLOPE - ENERGY OR RIVER BED SLOPE
C       TEMPR - WATER TEMPERATURE
C       VOL   - VOLUME
C       VSET  - SETTLING VELOCITY
C
C   CALLED BY: SAND
C
      REAL K4FUNC,K4
C
      DIMENSION VSET(3)
C
      CONST1 = 3.7975E-5
      CONST2 = 5.60249E+22
      CONST3 = 2.976328E+3
      FDIAM=D50 * 3.280833
      TMPR=TEMPR * 1.80 + 32.0
      V=(QTOT * ALEN / (VOL)) * CONST1
      FHRAD=HRAD * 3.280833
C
C   FOR WATER TEMPERATURES GREATER THAN 32F AND LESS THAN 100F
C   THE KINEMATIC VISCOSITY CAN BE WRITTEN AS THE FOLLOWING:
C
      VIS=4.106E-4 * (TMPR ** -0.864)
C
C   ASSUMING THE D50 GRAIN SIZE (DIAM) IS APPROXIMATELY
C   EQUAL TO THE GEOMETRIC MEAN GRAIN SIZE AND SIGMA-G = 1.5,
C   THE D65 GRAIN SIZE CAN BE DETERMINED TO BE 1.17 * D50.
C
      D65=1.17 * FDIAM
      CNV=0.1198 + 0.00048 * TMPR
      CZ=260.67 - 0.667 * TMPR
      TT=1.10 * (0.051 + 0.00009 * TMPR)
      ZI=VSET(1) * CONST1 * V / (CZ * FHRAD * SLOPE)
      IF(ZI.LT.CNV) ZI=1.5 * CNV
C
C   THE MANNING-STRICKLER EQUATION IS USED HERE TO
C   DETERMINE THE HYDRAULIC RADIUS COMPONENT DUE TO
C   GRAIN ROUGHNESS (R'), TAKEN FROM THE 1975 ASCE
C   "SEDIMENTATION ENGINEERING",PG. 128.
```

```
C
C      SUBSTITUTIONS ARE MADE IN EQUATION 2.141 FOR SHEAR VELOCITY
C      AND K(SUB)S.  THE FORMER IS REPLACED BY EQUATION 2.142, AND
C      THE LATTER BY D(SUB)65.
C
       RPRIME=((V**1.5) * (D65 **0.25) / (SLOPE ** 0.75)) * 0.00349
       USTAR=(RPRIME * SLOPE * 32.2) ** 0.5
       AFUNC=(VIS * 1.0E5) ** 0.333 / (10.0 * USTAR)
       CONDITIONAL
       .  (AFUNC .LE. 0.500)  AC = (AFUNC/4.89)** -1.45
       .  (AFUNC .LE. 0.660)  AC = (AFUNC/0.0036)**0.67
       .  (AFUNC .LE. 0.720)  AC = (AFUNC/0.29)**4.17
       .  (AFUNC .LE. 1.25)   AC = 48.0
       .  (AFUNC .GT. 1.25)   AC = (AFUNC/0.304)**2.74
       ...FIN
C
       K4FUNC=AFUNC * SLOPE * D65 * 1.0E5
       CONDITIONAL
       .  (K4FUNC.LE.0.25)  K4 = 1.0
       .  (K4FUNC.LE.0.35)  K4 = (K4FUNC**1.10) * 4.81
       .  (K4FUNC.GT.0.35)  K4 = (K4FUNC** -1.05) * 0.49
       ...FIN
C
       ACK4=AC * K4
       IF (ACK4 .LT. 16.0)
       .  ACK4=16.0
       .  K4=16.0 / AC
       ...FIN
       OCZU=1.0 + CNV - 1.5 * ZI
       OCZM=1.0 + CNV - ZI
       OCZL=1.0 + CNV - 0.756 * ZI
       ZINV=CNV - 0.756 * ZI
       ZM=-ZINV
       ZN=1.0 + ZINV
       ZO=-0.756 * ZI
       ZP=0.244 * ZI
       ZQ=0.5 * ZI
C
C    CLI HAS BEEN MULTIPLIED BY 1.0E30 TO KEEP IT FROM
C    EXCEEDING THE COMPUTER OVERFLOW LIMIT
C
        CLI=CONST2 * OCZL * (V ** 2.333) / FHRAD ** (ZM) /
      1     ((TT * AC * K4 * FDIAM) ** 1.667) / (1.0 + CNV) /
      2     ((FHRAD / 11.24) ** (ZN) - (2.0 * FDIAM) ** OCZL)
C
        P1=(2.0*FDIAM/FHRAD)**ZO
        C2D=CLI * P1 / 1.0E+30
C
C    CHECK TO SEE IF THE CALCULATED VALUE IS REASONABLE
C    (< 100.0), AND ADJUST IT IF IT IS NOT.
C
        IF(C2D.GT.100.0) CLI=  1.0E+28  / P1
C
C
C    CMI HAS BEEN MULTIPLIED BY 1.0E30 TO KEEP IT FROM
C    EXCEEDING THE COMPUTER OVERFLOW LIMIT
```

```
C
C      P(SUB)I APPEARING IN EQUATIONS 2,236H, J, K, L, M, AND
C        N IS THE WEIGHT FRACTION OF TOTAL SAND THAT THE I-TH
C        SIZE FRACTION CONTAINS.  SINCE WE ARE MODELING ALL
C        SAND AS A SINGLE SIZE FRACTION -- P(SUB)I = 1,0, AND
C        HENCE DOES NOT APPEAR IN THE MODEL EQUATIONS.
C
       CMI=43,2 * CLI * (1,0 + CNV) * V * (FHRAD ** (ZM))
C
C    CALCULATE TRANSPORT CAPACITY OF THE UPPER LAYER
C
       FD11=FHRAD / 11,24
       FD25=FHRAD / 2,5
       GSU=(CMI * (FD11 ** (ZP)) * (FD25 ** (ZQ)) *
      1    (FHRAD ** (OCZU) - (FD25 ** (OCZU)))) / (OCZU * 1,0 E+30)
C
C    CALCULATE THE CAPACITY OF THE MIDDLE LAYER.
C
       GSM=(CMI * (FD11 ** (ZP)) * ((FD25 **(OCZM)) -
      1    (FD11 ** (OCZM)))) / (OCZM * 1,0E+30)
C
C    CALCULATE THE CAPACITY OF THE LOWER LAYER
C
       GSL=(CMI * ((FD11 ** (ZN)) - ((2,0 * FDIAM) ** (OCZL))))
      1    / (OCZL * 1,0E+30)
C
C    CALCULATE THE CAPACITY OF THE BED LAYER
C
       GSB=(CMI * ((2,0 * FDIAM) ** (ZN)))/1,0E+30
C
C    TOTAL CAPACITY OF THE SEGMENT (GSI HAS UNITS OF TONS/DAY/FT)
C
       GSI=GSU + GSM + GSL + GSB
C
C    CONVERTING  TO KG/DAY/M
C
       GSU = GSU * CONST3
       GSM = GSM * CONST3
       GSL = GSL * CONST3
       GSB = GSB * CONST3
       YU = HRAD / 2,5
       YM = HRAD / 11,24
       YL = 2,0 * D50
       GSI=GSI * CONST3
C
       RETURN
       END

    (FLECS VERSION 22,46)

    ---------------------------------------
```

```
      ---------------------------------------
      SUBROUTINE TRANSP(FERROR,PCOEF,BWID,AWID,ABAR,DEPTH,OLDC,ECHO4,
     1                  CROSEC,ECHO7, ECHO8, ECHO9)
C
C     THIS ROUTINE SOLVES THE MASS TRANPORT EQUATIONS BY AN IMPLICIT
C     FINITE-ELEMENT METHOD.  A CRANK-NICHOLSON METHOD IS USED TO
C     APPROXIMATE THE SOLUTION THROUGH TIME.
C
C     VARIABLE DEFINITIONS:
C       ALEN   - SEGMENT LENGTH
C       AREA   - ELEMENT AREAS
C       B      - BED CONDITIONS
C       BDIV   - STANDARD BED LAYER THICKNESS
C       BED    - BED THICKNESS
C       C      - WATER CONDITIONS
C       CCIN   - CONCENTRATION OF INFLOW
C       COLD   - CELL-CENTERED CONCENTRATION
C       CROSEC - TOTAL CROSS-SECTIONAL AREA, M**2
C       DECAY  - FIRST ORDER DECAY
C       DELTH  - TIME STEP (SECONDS)
C       DELZ   - STANDARD ELEMENT THICKNESS
C       DENS   - DENSITY
C       DFZ    - DIFFUSION COEFFICIENT
C       DIAM   - DIAMETER
C       DSHR   - CRITICAL SHEAR STRESS FOR DEPOSITION
C       D50    - MEDIAN BED SEDIMENT DIAMETER (M)
C       ERODE  - ERODABILITY
C       FERROR - FATAL ERROR FLAG (L*1)
C       HRAD   - HYDRAULIC RADIUS
C       KAY1   - LIGHT EXTINCTION COEFFICIENT OF WATER
C       KAY2   - LIGHT EXTINCTION OF SUSPENDED SEDIMENT IN WATER
C       NBED   - NUMBER OF BED LAYERS
C       NELEM  - NUMBER OF ELEMENTS
C       PCOEF  - FIRST TERM OF THE PHOTOLYSIS RATE OF CHANGE EQUATION
C       POR    - POROSITY
C       QHIN   - INFLOW DISCHARGE
C       QHOUT  - OUTFLOW DISCHARGE
C       QV     - VERTICAL DISCHARGE
C       SCSHR  - CRITICAL SHEAR STRESS FOR SCOUR
C       SLOPE  - ENERGY OR RIVER BED SLOPE
C       SMETH  - CONTROL VARIABLE TO SELECT THE METHOD TO BE USED
C                WHEN COMPUTING THE SAND CARRYING CAPACITY. (BYTE)
C       SR     - SEDIMENT EROSION RATE
C       STRESS - BED SHEAR STRESS
C       TEMPR  - WATER TEMPERATURE
C       VOL    - VOLUME
C       VSET   - PARTICLE SETTLING VELOCITY
C       XYSD   - THICKNESS OF TOP BED LAYER
C
C     CALLED BY: SERATRA
C     CALLS: BEDOK, BEDHIS, COLAPS, COMB, DISOLV, FALL, PARTIC, SAND,
C            SEDIME, SETUP, SILCAL, TRISOL
C
      INCLUDE 'ELMSIZ.PRM'
```

```
      C
            LOGICAL*1 FERROR, ECHO4, ECHO7, ECHO8, ECHO9
      C
      C
            DIMENSION A(MXELEM), ABAR(MXELEM), OD(10), ILAYR(3), P(MXELEM,3),
           1          R(MXELEM), S(MXELEM,3), SD(MXELEM,6),
           2          SR(6), XNT(3), Z(MXELEM), BWID(MXELEM), AWID(MXELEM)
           3    ,OLDC(MXELEM,MAXCON), DEPO(6), SCOUR(6), BEDSD(3)
      C
            INCLUDE 'TRANS.COM'
      C
      C
            DATA EPSI/1.0E-30/
      C
            MP1 = NELEM + 1
            ML1 = NELEM - 1
      C
      C     *** PERFORM CALCULATIONS OVER THE TIME STEP DELTD ***
      C
            DELTD = DELTH / 86400.
            DO (J=1,MAXCON)
            .  DO (L = 1,MP1)
            .  .  P(L) = 0.0
            .  .  DO (N = 1,3)
            .  .  .  S(L,N) = 0.0
            .  .  .  P(L,N) = 0.0
            .  .  ...FIN
            .  ...FIN
      C     .
      C     .
            .  DO (I=1,NELEM)
            .  .  RCDZ = 0.0
            .  .  IF (I .EQ. 1)
            .  .  .  SELECT(J)
            .  .  .  .  (1)
            .  .  .  .  .  CALL SAND (ABAR, ALEN, AREA, B, BDIV, CCIN, DELTD, DELZ,
           1.  .  .  .  .             DENS, D50, HRAD, NBED, NELEM, POR, QHIN, QHOUT,
           2.  .  .  .  .             SCSHR, SLOPE, SMETH, STRESS, TEMPR, VSET,VOL, XYSO,
           3.  .  .  .  .             DEPO,ILAYR, SD, SR, XNT, COLD, C,
           4.  .  .  .  .             CRUSEC,BWID,ECHO7, SCOUR)
            .  .  .  .  .  IF(ECHO8)
            .  .  .  .  .  .  WRITE(6,1000)
            .  .  .  .  .  .  WHEN(ILAYR(1).LT.0)
            .  .  .  .  .  .  .  WRITE(6,1010) J,ILAYR(1),DEPO(1),DEPO(4),SD(1,1),XNT(1),
           1.  .  .  .  .  .  .           (SD(II,4),II=1,NELEM)
            .  .  .  .  .  .  ...FIN
            .  .  .  .  .  .  ELSE
            .  .  .  .  .  .  .  WRITE(6,1015) J,ILAYR(1),SCOUR(1),SR(1),XNT(1),SCOUR(4),SR(4
            .  .  .  .  .  .  ...FIN
       1010 .  .  .  .  .  .    FORMAT(' J=',I2,'  ILAYR=',I2,'  DEPO=',E15.7,'  DEPO(+3)=',
           1.  .  .  .  .  .  E15.7,'  SD=',E15.7,'  XNT=',E15.7,5X,'SD(I,+3)='/
           2.  .  .  .  .  .  (5X,8E15.7))
       1015 .  .  .  .  .  .  FORMAT(' J=',I2,'  ILAYR=',I2,'  SCOUR=',E15.7,'  SR=',E15.7,
           1.  .  .  .  .  .  '  XNT=',E15.7,'  SCOUR(+3)=',E15.7,'  SR(+3)=',E15.7)
            .  .  .  .  .  ...FIN
       1000 .  .  .  .  .  FORMAT('0IN TRANSP FOLLOWING SAND')
```

```
           .  .  .  .  ...FIN
           .  .  .  .  (2)
           .  .  .  .  .  CALL SILCLA(ABAR, B, BDIV, CCIN, DELTD, DELZ, DEPTH, DENS,
          1.  .  .  .  .              DSHR, ERODE, HRAD, 2, NBED,COLD,NELEM, POR, QHIN,
          2.  .  .  .  .              QHOUT, SCSHR, STRESS, VSET, XYSO, DEPO, ILAYR,
          3.  .  .  .  .              SD, SR, XNT, CROSEC, BWID, ALEN, ECHO7, SCOUR)
           .  .  .  .  .  IF(ECHO8)
           .  .  .  .  .  .  WRITE(6,1020)
           .  .  .  .  .  .  WHEN(ILAYR(2).LT.0)
           .  .  .  .  .  .  .  WRITE(6,1010) J,ILAYR(2),DEPO(2),DEPO(5),SD(1,2),XNT(2),
          1.  .  .  .  .  .  .              (SD(II,5),II=1,NELEM)
           .  .  .  .  .  .  ...FIN
           .  .  .  .  .  .  ELSE
           .  .  .  .  .  .  .  WRITE(6,1015) J,ILAYR(2),SCOUR(2),SR(2),XNT(2),SCOUR(5),SR(5)
           .  .  .  .  .  .  ...FIN
           .  .  .  .  .  ...FIN
   1020    .  .  .  .  .  FORMAT('0IN TRANSP FOLLOWING SILT')
           .  .  .  .  ...FIN
           .  .  .  .  (3)
           .  .  .  .  .  CALL SILCLA(ABAR, B, BDIV, CCIN, DELTD, DELZ, DEPTH, DENS,
          1.  .  .  .  .              DSHR, ERODE, HRAD, 3, NBED,COLD,NELEM, POR, QHIN,
          2.  .  .  .  .              QHOUT, SCSHR, STRESS, VSET, XYSO, DEPO, ILAYR,
          3.  .  .  .  .              SD, SR, XNT, CROSEC, BWID, ALEN, ECHO7, SCOUR)
           .  .  .  .  .  IF(ECHO8)
           .  .  .  .  .  .  WRITE(6,1030)
           .  .  .  .  .  .  WHEN(ILAYR(3).LT.0)
           .  .  .  .  .  .  .  WRITE(6,1010) J,ILAYR(3),DEPO(3),DEPO(6),SD(1,3),XNT(3),
          1.  .  .  .  .  .  .              (SD(II,6),II=1,NELEM)
           .  .  .  .  .  .  ...FIN
           .  .  .  .  .  .  ELSE
           .  .  .  .  .  .  .  WRITE(6,1015) J,ILAYR(3),SCOUR(3),SR(3),XNT(3),SCOUR(6),SR(6)
           .  .  .  .  .  .  ...FIN
           .  .  .  .  .  ...FIN
   1030    .  .  .  .  .  FORMAT('0IN TRANSP FOLLOWING (SILT) CLAY')
           .  .  .  .  ...FIN
           .  .  .  ...FIN
           .  .  ...FIN
           .  .  CONDITIONAL
           .  .  .  (J .LE. 3)
           .  .  .  .  CALL SEDIME(ABAR, ALEN, CCIN, DELTD, I, J, NELEM, QHIN,
          1.  .  .  .            QHOUT, QV, SD, SR, ALFA, BETA, VEL1, VEL2,VSET,
          2.  .  .  .        DELZ, BWID, AWID, DEPTH, BETA1, BETA2)
           .  .  .  .  IF(ECHO8) WRITE(6,1500)
   1500    .  .  .  .  FORMAT('0IN TRANSP FOLLOWING SEDIME')
           .  .  .  ...FIN
           .  .  .  (J .GE. 4 .AND. J .LE. 6)
           .  .  .  .  CALL PARTIC(ABAR, ALEN, B, C, CCIN, COLD, DECAY, DELTD, DFZ,
          1.  .  .  .    I, J, NBED, NELEM, QHOUT, QHIN, QV, SORBK, SD, SR, ALFA, BETA,
          2.  .  .  .            VEL1, VEL2,VSET, DELZ, DEPTH,BWID,AWID,
          3.  .  .  .            BETA1, BETA2)
           .  .  .  .  IF(ECHO8) WRITE(6,1510)
   1510    .  .  .  .  FORMAT('0IN TRANSP FOLLOWING PARTIC')
           .  .  .  ...FIN
           .  .  .  (J .EQ. 7)
           .  .  .  .  CALL DISOLV(ABAR,  B, BDIV, C, CCIN, COLD, DECAY, DELZ,
          1.  .  .  .            DELTD, DENS, DIAM,  I, KAY1, KAY2, NELEM,NBED,
```

```
            2.  .  .  .               PCOEF, POR, QHIN, QHOUT, QV, SORBK, ALFA, BETA, VEL1,
            3.  .  .  .               VEL2, BETA1, BETA2, DEPO, SCOUR, BEDSD, XYSD, AREA)
                .  .  .  .  IF(ECHO8) WRITE(6,1520) (BEDSD(II),II=1,3)
   1520         .  .  .  .  FORMAT('0IN TRANSP FOLLOWING DISOLV        BEDSD(1-3) =',3E18.7)
                .  .  .  ...FIN
                .  .  ...FIN
                .  .  IF(ECHO8) WRITE(6,1530) VEL1,VEL2
   1530         .  .  FORMAT('  VEL1 =',E15.7,10X,'VEL2 =',E15.7)
   C            .  .
   C            .  .  *** CONSTRUCT THE FINITE ELEMENT MATRICES FOR EACH LAYER ***
                .  .  WHEN (J.LE.3) DIFUSE = DFZ(J)
                .  .  ELSE DIFUSE = DFZ(J-3)
                .  .  DD(1) = DIFUSE/(DELZ*DELZ)
                .  .  DD(2) = 6.0 * DELZ
                .  .  DD(3) = (VEL2 - 4. * VEL1) / DD(2)
                .  .  DD(4) = (VEL1 + 2. * VEL2) / DD(2)
                .  .  DD(5) = (2. * VEL1 + VEL2) / DD(2)
                .  .  DD(6) = (4. * VEL2 - VEL1) / DD(2)
                .  .  DD(7) = ALFA
                .  .  DD(8) = BETA
                .  .  DD(9) = BETA1
                .  .  DD(10) = BETA2
                .  .  IF (ECHO4)
                .  .  .  WRITE(6,2500)
   2500         .  .  .  FORMAT('0  I',1X,'J',6X,'DD(1)',7X,'DD(2)',7X,'DD(3)',7X,'DD(4)',7
            1.  .  .  X,'DD(5)',7X,'DD(6)',7X,'DD(7)',7X,'DD(8)',7X,'DD(9)',7X,'DD(10)')
                .  .  .  WRITE(6,2000)I,J,(DD(K),K=1,10)
   2000         .  .  .  FORMAT(1X,I3,1X,I1,1X,10(1PE12.4))
                .  .  ...FIN
   C            .  .
                .  .  VEL = VEL1/DELZ
                .  .  IF(I.EQ.NELEM) VEL=-VEL2/DELZ
                .  .  WIDTH = AWID(I)
                .  .  CALL SETUP(I, DD, P, S, R, VEL,NELEM,ECHO4,WIDTH)
                .  ...FIN
   C            .
                .  HALFD = DELTD / 2.0
                .  DO (L = 1,MP1)
                .  .  R(L) = R(L) + DELTD
                .  .  Z(L) = C(L,J)
                .  .  DO (N = 1,3)
                .  .  .  P1 = S(L,N)*HALFD
                .  .  .  PBAR = P(L,N) + P1
                .  .  .  S(L,N) = P(L,N) - P1
                .  .  .  P(L,N) = PBAR
                .  .  ...FIN
                .  ...FIN
   C            .
                .  IF (ECHO4)
                .  .  WRITE(6,5760)
   5760         .  .  FORMAT(' BEFORE COMB************BEFORE COMB**')
                .  .  WRITE(6,5761)
   5761         .  .  FORMAT('0NODE',6X,'P(I,1)',10X,'P(I,2)',10X,'P(I,3)',10X,
            1.  .  'S(I,1)',10X,'S(I,2)',10X,'S(I,3)',11X,'Z(I)',12X,'R(I)')
                .  .  WRITE(6,5762)(I,(P(I,K),K=1,3),(S(I,K),K=1,3),Z(I),R(I),
            1.  .                 I=1,MP1)
```

```
00222    5762  .  .  FORMAT(1X,I2,1P8E16.5)
00223          .  ...FIN
00224          .  CALL COMB(MP1, S, Z, R)
00225          .  IF (ECHO4) WRITE(6,6000)(R(L),L=1,MP1)
00226     6000 .  FORMAT(' R  AFTER COMB',8(1PE12.4,2X))
00227    C     .
00228    C     .  *** SOLVE THE SYSTEM OF EQUATIONS BY GAUSSIAN ELIMINATION ***
00229    C     .
00230          .  CALL TRISOL(MP1,P(2,1), P(1,2), P(1,3),R)
00231    C     .
00232          .  DO (I=1,MP1)
00233          .  .  OLDC(I,J)=C(I,J)
00234          .  .  C(I,J)=R(I)
00235          .  ...FIN
00236          .  COMPUTE-CELL-CENTERED-VALUES
00237    C     .
00238          ...FIN
00239          CALL BEDHIS(B, BDIV, BED, COLD, DELTD, DELZ,  DENS,
00240         1              FERROR, ILAYR, NBED, NELEM, POR, XNT, XYSO,
00241         2              DEPO, SCOUR, BEDSD)
00242          IF(DECAY(1).GT.0.0) CALL BEDDK(B,DECAY,DELTD,NBED)
00243          RETURN

         ----------------------------------------

00244          TO COMPUTE-CELL-CENTERED-VALUES
00245    C     .
00246    C     .  *** COLLAPSE THE NODAL VALUES OF C INTO CELL CENTERED VALUES
00247    C     .      IN COLD ***
00248    C     .
00249          .  DO (I=1,NELEM)
00250          .  .  COLD(I,J) = (C(I,J) + C(I+1,J))/2.
00251          .  ...FIN
00252     6200 .  FORMAT(' C IN PROCEDURE I=',I2,2X,1P7E14.4)
00253          .  IF (ECHO4)
00254          .  .  IF(J.EQ.MAXCON)
00255          .  .  .  WRITE(6,6200)(I,(C(I,JJ),JJ=1,MAXCON),I=1,MP1)
00256          .  .  .  WRITE(6,6300)(I,(COLD(I,JJ),JJ=1,MAXCON),I=1,NELEM)
00257          .  .  ...FIN
00258          .  ...FIN
00259     6300 .  FORMAT(' COLD IN PROCEDRE I=',I2,2X,1P7E14.4)
00260          ...FIN
00261          END

         ----------------------------------------

               PROCEDURE CROSS-REFERENCE TABLE

         00244 COMPUTE-CELL-CENTERED-VALUES
               00236


         (FLECS VERSION 22.46)

         ----------------------------------------
```

```
        ------------------------------------

          SUBROUTINE TRBDAT(ECHO, HLDERR, NELEM, NTRIBS, NUMERR,
         1                      SIMLEN, TRBOPT)
C
C    THIS ROUTINE IS RESPONSIBLE FOR READING AND PROCESSING THE
C    TRIBUTARY INFLOW MASS FLUX DATA.  THE DATA IS READ FROM
C    THE INPUT STREAM (LUN 1) AND WRITTEN TO "TRIBUTARY.TMP" (LUN 7)
C    FOR USE DURING THE SIMULATION.
C
C      FORMAL PARAMETERS:
C        ECHO   - LINE PRINTER ECHO OPTION CONTROL VARIABLE (L*1)
C        HLDERR - HOLDING ARRAY FOR ERROR NUMBERS (BYTE)
C        NELEM  - NUMBER OF VERTICAL ELEMENTS
C               NOTE: NELEM IS LATER REDEFINED IN HYDDAT
C        NTRIBS - NUMBER OF TRIBUTARIES (0 OR 1)
C        NUMERR - NUMBER OF INPUT ERRORS
C        SIMLEN - SIMULATION LENGTH (SECONDS - I*4)
C        TRBOPT - TRIBUTARY INPUT CONTROL VARIABLE
C
C    CALLED BY: SERATRA
C    CALLS: PUTERR
C
          INCLUDE 'SY:ELMSIZ.PRM'
C
          BYTE HLDERR(100)
C
          INTEGER*2 TRBOPT
C
          INTEGER*4 ENDTIM,PRETIM,SIMLEN
C
          LOGICAL*1 ECHO
C
          DIMENSION CTRB(MXELEM,MAXCON)
C
C    ....TRIBUTARY INFLOW MASS FLUX
C
C        FIRST RECORD.....
C
C            COL.  1- 5...NTRIBS....NUMBER OF TRIBUTARIES (0 OR 1)
C                  6-10...TRBOPT....TRIBUTARY INPUT OPTION
C                                   =0: THE USER WANTS THE MODEL TO
C                                       DISTRIBUTE THE MASS FLUX THRU
C                                       THE ELEMENTS.
C                                   =1: THE USER WILL SUPPLY THE
C                                       MASS FLUX VALUES FOR EACH ELEMENT
C
          REWIND 7
C
          READ(1,2) NTRIBS,TRBOPT
C
          IF (ECHO) WRITE(6,3) NTRIBS,TRBOPT
C
          IF (NTRIBS .GT. 0)
          .  IF(ECHO) WRITE(6,4)
```

```
                . REPEAT UNTIL (ENDTIM .EQ. -9999)
      C         . .
      C         . .  CARD 11-8
      C         . .  COL.  1-10...ENDTIM....ENDING TIME FOR THE DATA THAT FOLLOWS,
      C         . .                          A VALUE OF -9999 TERMINATES THE DATA
      C         . .                             (SECONDS).
      C         . .
                . .  READ(1,6) ENDTIM
      C         . .
                . .  UNLESS (ENDTIM .EQ. -9999)
      C         . .  .
      C         . .  .  RECORD TWO.......TRIBUTARY MASS FLUX AND DEPTH
      C         . .  .            ********** CAUTION **********
      C         . .  .            THE MASS FLUX UNITS ARE DIFFERENT FROM THOSE OF
      C         . .  .            INITIAL WATER AND UPSTREAM WATER CONCENTRATIONS.
      C         . .  .            RADIONUCLIDE IS PC/SEC, PESTICIDE IS KG/SEC,
      C         . .  .            SEDIMENT IS KG/SEC
      C         . .  .            ********** CAUTION **********
      C         . .  .
      C         . .  .  COL.  1-10...CTRB(1,1)...MASS FLUX OF SAND (KG/M**3)*(M**3/SEC)
      C         . .  .       11-20...CTRB(1,2)...MASS FLUX OF SILT
      C         . .  .       21-30...CTRB(1,3)...MASS FLUX OF CLAY
      C         . .  .       31-40...CTRB(J,4)...MASS FLUX OF CONTAMINANT ASSOCIATED
      C         . .  .                           WITH SAND (PC/KG)*(KG/M**3)*(M**3/SEC)
      C         . .  .       41-50...CTRB(J,5)...MASS FLUX OF CONTAMINANT ASSOCIATED
      C         . .  .                           WITH SILT
      C         . .  .       51-60...CTRB(J,6)...MASS FLUX OF CONTAMINANT ASSOCIATED
      C         . .  .                           WITH CLAY
      C         . .  .       61-70...CTRB(J,7)...MASS FLUX OF DISSOLVED CONTAMINANT
      C         . .  .                       (PC/M**3)*(M**3/SEC)
      C         . .  .
                . .  .  WHEN (TRBOPT .EQ. 0) N = 1
                . .  .  ELSE N = NELEM
                . .  .  DO (J=1,N)
                . .  .  .  READ(1,1) (CTRB(J,I),I=1,MAXCON)
                . .  .  .  IF (ECHO)
                . .  .  .  .  WRITE(6,5) ENDTIM,(CTRB(J,I),I=1,MAXCON)
                . .  .  .  ...FIN
                . .  .  ...FIN
                . .  .  WRITE(7) ENDTIM,((CTRB(J,I),I=1,7),J=1,N)
                . .  .  PRETIM = ENDTIM
                . .  ...FIN
                . ...FIN
                . REWIND 7
                . IF (PRETIM .LT. SIMLEN) CALL PUTERR(23, NUMERR, HLDERR)
                ...FIN
      C
                RETURN
      C
        1       FORMAT(7F10.0)
        2       FORMAT(2I5)
        3       FORMAT(1H0,4X,'TRIBUTARY DATA'/14X,I5,'...NUMBER OF TRIBUTARIES'
               1  /14X,I5,'...TRIBUTARY INPUT CONTROL VARIABLE')
        4       FORMAT(1H0,'ENDING TIME',1X,3(3X,'CONC. OF',4X),3(1X,'CONC.'
               1  ' ASSOC.',2X),2X,'CONTAMINANT',7X,'FLOW'/13X,'SUSPENDED SAND'
               2  ,1X, 'SUSPENDED SILT',1X,'SUSPENDED CLAY',3X,'WITH SAND',6X,
```

```
          3  'WITH SILT',6X,'WITH CLAY',4X,'DISSOLVED CONC',2X,
          4  '(M**3/SEC)')
      5    FORMAT(2X,I10,2X,7(1PE12.5,3X),1PE12.5)
      6    FORMAT(I10)
C
           END
```

(FLECS VERSION 22,46)

```
----------------------------------------

      SUBROUTINE TRBFLO(CTRB, CTRIB, ENDTRB, ETIME, FERROR, DEPTH,
     1                 NELEM, NEWQI, NEWTRB, QHIN, TRBOPT, DEPMIN)
C
C       WHEN THERE IS A TRIBUTARY TO THE SEGMENT THIS SUBROUTINE IS
C       CALLED EACH TIME STEP TO READ THE DATA FROM LUN 7
C       WHICH WAS WRITTEN BY SUBROUTINE TRBDAT.
C
C       FORMAL PARAMETERS:
C          CTRB   - REDISTRIBUTED CONCENTRATIONS
C          CTRIB  - ORIGINAL TRIBUTARY MASS FLUX
C          ENDTRB - ENDING TIME OF THE CURRENT TRIBUTARY DATA (I*4)
C          ETIME  - ELAPSED TIME OF THE SIMULATION (I*4)
C          FERROR - FATAL ERROR FLAG
C          NELEM  - NUMBER OF ELEMENTS
C                CAUTION: NELEM HAS BEEN REDEFINED IN HYDDAT
C                SINCE ITS USE IN TRBDAT
C          NEWQI  - NEW QHIN DATA FLAG (L*1)
C          NEWTRB - NEW TRIBUTARY DATA FLAG (L*1)
C          QHIN   - INFLOW TO CURRENT SEGMENT
C          TRBOPT - TRIBUTARY INPUT OPTION CONTROL VARIABLE (I*2)
C
C       CALLED BY: SERATRA
C
      INCLUDE 'SY:ELMSIZ.PRM'
C
      INTEGER*2 TRBOPT
C
      INTEGER*4 ENDTRB,ETIME
C
      LOGICAL*1 NEWTRB,NEWQI,FERROR
C
      DIMENSION CTRB(MXELEM,MAXCON), CTRIB(MAXCON), QHIN(MXELEM)
C
      NEWTRB = .FALSE.
      IF (ETIME .GT. ENDTRB)
      .  NEWTRB = .TRUE.
      .  REPEAT UNTIL(ETIME .LE. ENDTRB)
  100 .  .  CONTINUE
      .  .  WHEN (TRBOPT .EQ. 0)
      .  .  .  READ(7,END=200) ENDTRB, (CTRIB(J),J=1,MAXCON)
      .  .  ...FIN
      .  .  ELSE READ(7,END=200) ENDTRB,((CTRB(I,J),J=1,MAXCON),I=1,NELEM)
      .  ...FIN
      ...FIN
      IF (DEPTH .LE. DEPMIN) RETURN
      IF (TRBOPT .EQ. 0 .AND. (NEWTRB .OR. NEWQI))
      .  QHTOT = 0.0
      .  DO (I=1,NELEM) QHTOT = QHTOT + QHIN(I)
C     .  *** DISTRIBUTE THE MASS FLUX THROUGHOUT THE ELEMENTS ***
C     .   NOTE: UNITS ARE
C     .                    SEDIMENT KG/SEC
C     .                    PARTICULATE  PC/SEC OR KG/SEC
C     .                    DISSOLVED    PC/SEC
```

```
      .  DO (K=1,MAXCON)
      .  .  DO (I=1,NELEM)
      .  .  .  CTRB(I,K) = CTRIB(K) * QHIN(I) / QHTOT
      .  .  ...FIN
      .  ...FIN
      ...FIN
      RETURN
C
 200  CONTINUE
      FERROR = .TRUE.
      WRITE(6,1)
1     FORMAT(10X,'FATAL ERROR - TRIBUTARY DATA EXHAUSTED')
      RETURN
C
      END

(FLECS VERSION 22.46)

----------------------------------------
```

(FLECS VERSION 22.46)   13-MAR-81   13:32:36 PAGE 00001

```
---------------------------------------

        SUBROUTINE TRISOL(MP1,D1,D,D2,R)
C       CALLED BY TRANSP.
C       USED IN ORIGINAL VERSION OF SERATRA
        INCLUDE 'ELMSIZ.PRM'
C
        DIMENSION D(MXELEM), D1(MXELEM), D2(MXELEM), R(MXELEM)
C
        N=MP1
        N1=N-1
C
C       FORWARD ELIMINATION
C
        DO(I=1,N1)
        .  D1D=D1(I)/D(I)
        .  D(I+1)=D(I+1)-D2(I)*D1D
        .  R(I+1)=R(I+1)-R(I)*D1D
        ...FIN
C
C       BACKWARD SUBSTITUTION
C
        R(N)=R(N)/D(N)
        DO (I=1,N1)
        .  K=N-I
        .  R(K)=(R(K)-D2(K)*R(K+1))/D(K)
        ...FIN
C
        RETURN
        END

    (FLECS VERSION 22.46)

---------------------------------------
```

```
      ------------------------------------------
      SUBROUTINE TRNPOS(ABAR,AREA,AWID,ALEN,BWID,DELZ,EL,
     1          IELM,NELEM,XSAREA,VOL)
C
C     THIS SUBROUTINE CONSERVE THE GROSS CROSS-SECTINAL AREA  AS A FUNCTION
C     OF DEPTH DURING THE CONVERSION OF THE REAL CROSS-SECTION TO ITS
C     IDEALIZED RECTILINEAR SHAPE.
C
      INCLUDE 'SY:ELMSIZ.PRM'
C
      DIMENSION ABAR(MXELEM), AREA(MXELEM), AWID(MXELEM),
     1         BWID(MXELEM), EL(MXELEM), IELM(MXELEM),
     2         XSAREA(MXELEM)
C
C
      BLEN=1./ALEN
      ELTOP=DELZ
      WB=AREA(1)*BLEN
      BWID(1)=WB
      ARAB=(WB+AREA(2)*BLEN)*(EL(2)-EL(1))/2.
      ARRB=0.
      NB=1
C
C     DETERMINE LOCATION OF TOP NODE WITH RESPECT TO ORIGINAL DATA
C
      DO(I=1,NELEM)
      .  DO(J=1,MXELEM-1)
      .  .  IF(ELTOP .GE. EL(J) .AND. ELTOP .LE. EL(J+1))
      .  .  .  EB=EL(J)
      .  .  .  ET=EL(J+1)
      .  .  .  WB=AREA(J)*BLEN
      .  .  .  WT=AREA(J+1)*BLEN
      .  .  .  NT=J
      .  .  .  IELM(I)=J
      .  .  .  GO TO 10
      .  .  ...FIN
      .  ...FIN
   10 .  CONTINUE
C     .
C     .  LINEARLY INTERPOLATE WIDTH AT ELEMENT'S TOP NODE
C     .
      .  WTOP=WB + (ELTOP-EB)*(WT-WB)/(ET-EB)
C     .
C     .  ASSUME TRAPZOIDAL SHAPES TO FIND CROSS-SECTIONAL AREAS
C     .
      .  ARAT=(WT+WTOP)*(ET-ELTOP)/2.
      .  ARBT=(WTOP+WB)*(ELTOP-EB)/2.
C     .
C     .  DETERMINE IF NEW ELEMENT SURFACES HAVE BEEN FOUND TO
C     .  (A) LIE WITHIN A SINGLE DATA SET
C     .  (B) LIE IN SEQUENTIAL DATA SETS, OR
C     .  (C) BE SEPARATED BY ONE OR MORE DATA SETS
C     .  FINALLY, FORM THE CROSS-SECTIONAL AREA
C     .
```

```
          .  INDIC=NT-NB
          .  IF(INDIC .EQ. 0) XSAREA(I)=ARBT-ARBB
          .  IF (INDIC .GE. 1) XSAREA(I)=ARBT+ARAB
          .  IF (INDIC .GE. 2)
          .  .  XSAR=0.
          .  .  DO(II=NB+1,NT-1)
          .  .  .  XSAR=XSAR+(AREA(II)+AREA(II+1))*(EL(II+1)-EL(II))*BLEN/2.
          .  .  ...FIN
          .  .  XSAREA(I)=XSAREA(I)+XSAR
          .  ...FIN
C         .
C         .  DETERMINE AVERAGE VERTICAL PROJECTION, AVERAGE WIDTH, REAL WIDTH,VOLUM
C         .
          .  AWID(I)=XSAREA(I)/DELZ
          .  ABAR(I)=AWID(I)*ALEN
          .  BWID(I+1)=WTOP
          .  VOL=VOL+XSAREA(I)*ALEN
C         .
C         .  OVERWRITE INITIAL INFORMATION FOR NEXT ELEMENT
C         .
          .  ELTOP=ELTOP+DELZ
          .  ARAB=ARAT
          .  ARBB=ARBT
          .  NB=NT
          ...FIN
          RETURN
          END

     (FLECS VERSION 22,46)

     --------------------------------------
```

```
      ----------------------------------------
      SUBROUTINE UPSDAT(ECHO, HLDERR, NUMERR, SIMLEN,UWID,UEL)
C
C   THIS ROUTINE IS RESPONSIBLE FOR READING THE UPSTREAM INFLOW
C   CONDITIONS TO SEGMENT 1. THE DATA IS READ FROM THE INPUT STREAM
C   (LUN 1) AND THEN WRITTEN TO "DUMMY.DT1' (LUN 2) FOR SUBSEQUENT
C   USE DURING THE SIMULATION.
C
C     FORMAL PARAMETERS:
C       ECHO   - LINE PRINTER ECHO OPTION CONTROL VARIABLE (L*1)
C       HLDERR - HOLDING ARRAY FOR ERROR NUMBERS (BYTE)
C       NUMERR - NUMBER OF INPUT ERRORS
C       SIMLEN - SIMULATION LENGTH (SECONDS - I*4)
C
C   CALLED BY: SERATRA
C   CALLS: PUTERR
C
      INCLUDE 'ELMSIZ.PRM'
C
      BYTE HLDERR(100)
C
      INTEGER*4 ENDTIM,PRETIM,SIMLEN
C
      LOGICAL*1 ECHO
C
      DIMENSION CCIN(MXELEM,MAXCON), UWID(MXELEM), UEL(MXELEM)
C
      REWIND 2
C
      IF (ECHO) WRITE(6,6)
C
C     .....UPSTREAM INFLOW CONDITIONS TO SEGMENT 1
C
      REPEAT UNTIL (ENDTIM .EQ. -9999)
C     .
C     .    RECORD ONE.....
C     .     COL.  1-10...ENDTIM....ENDING TIME FOR DATA THAT FOLLOWS.
C     .                            A VALUE OF -9999 TERMINATES THE DATA.
C     .                            (SECONDS)
C     .          11-15...NM........NUMBER OF ELEMENTS
C     .        16-25...UDEPTH...ELEVATION OF FREE SURFACE ABOVE BED
C     .
      . READ(1,5) ENDTIM, NM, UDEPTH
C     .
      . IF(ENDTIM .NE. -9999)
C     . .
C     . .   RECORD TWO......WATER CONDITIONS. ONE CARD IS READ FOR EACH NODE.
C     . .                  THE UNITS OF THE CONTAMINANT CONCENTRATIONS DEPEND
C     . .                  UPON THE TYPE OF CONTAMINANT (RADIONUCLIDE..PC/KG
C     . .                  OR PESTICIDE..KG/KG). THE PARAMETERS ARE READ IN
C     . .                  THE FOLOWING ORDER:
C     . .
C     . .      PARAMETER 1...CONCENTRATION OF SAND (KG/M**3)
C     . .                2...CONCENTRATION OF SILT (KG/M**3)
```

```
C       .  .                    3...CONCENTRATION OF CLAY (KG/M**3)
C       .  .                    4...CONCENTRATION OF CONTAMINANT ASSOCIATED WITH
C       .  .                          SAND
C       .  .                    5...CONCENTRATION OF CONTAMINANT ASSOCIATED WITH
C       .  .                          SILT
C       .  .                    6...CONCENTRATION OF CONTAMINANT ASSOCIATED WITH
C       .  .                          CLAY
C       .  .                    7...CONCENTRATION OF DISSOLVED CONTAMINANT
C       .  .
        .  .  WHEN(NM .GT. 0)
        .  .  .  DO (I=1,NM)
        .  .  .  .  READ (1,1) (CCIN(I,K), K=1, MAXCON)
        .  .  .  ...FIN
        .  .  ...FIN
        .  .  ELSE
        .  .  .  NM=MXELEM
        .  .  .  READ (1,1) (CCIN(1,K), K=1, MAXCON)
        .  .  .  DO (I=2,NM)
        .  .  .  .  DO (K=1, MAXCON) CCIN(I,K)=CCIN(1,K)
        .  .  .  ...FIN
        .  .  ...FIN
        .  .  IF (ECHO)
        .  .  .  WRITE(6,4) ENDTIM
        .  .  .  WRITE(6,3)
        .  .  .  DO (I=1,NM)
        .  .  .  .  WRITE(6,2) I,(CCIN(I,K),K=1,MAXCON)
        .  .  .  ...FIN
        .  .  ...FIN
C       .  .
        .  .  WRITE(2) ENDTIM,NM,UDEPTH,((CCIN(I,K),K=1,MAXCON),I=1,NM)
        .  .  PRETIM = ENDTIM
        .  ...FIN
        ...FIN
        IF (PRETIM .LT. SIMLEN) CALL PUTERR(24, NUMERR, HLDERR)
C
        REWIND 2
C       RECORD THREE.....CHANNEL CROSS-SECTION DATA
C            .....UWID -- WIDTH OF SEGMENT AT NODES
C            .....UEL  -- ELEVATION OF NODE ABOVE BOTTOM
C
C*****************************************************************
C       CAUTION!
C       IF CONCENTRATIONS ARE INPUT AT VARIOUS DEPTHS,
C       THE CHANNEL CROSS-SECTIONAL DATA MUST BE INPUT
C       AT THOSE SAME DEPTHS.
C
C*****************************************************************
        READ(1,7) NWID, UWIDTH, DEL
        WHEN (NWID.EQ.0)
        .  E=0.
        .  DO (I=1,MXELEM)
        .  .  UWID(I)=UWIDTH
        .  .  UEL(I)=E
        .  .  E=E+DEL
        .  ...FIN
        ...FIN
```

```
 C     .  .                  3...CONCENTRATION OF CLAY (KG/M**3)
 C     .  .                  4...CONCENTRATION OF CONTAMINANT ASSOCIATED WITH
 C     .  .                        SAND
 C     .  .                  5...CONCENTRATION OF CONTAMINANT ASSOCIATED WITH
 C     .  .                        SILT
 C     .  .                  6...CONCENTRATION OF CONTAMINANT ASSOCIATED WITH
 C     .  .                        CLAY
 C     .  .                  7...CONCENTRATION OF DISSOLVED CONTAMINANT
 C     .  .
       .  .  WHEN(NM .GT. 0)
       .  .  .  DO (I=1,NM)
       .  .  .  .  READ (1,1) (CCIN(I,K), K=1, MAXCON)
       .  .  .  ...FIN
       .  .  ...FIN
       .  .  ELSE
       .  .  .  NM=MXELEM
       .  .  .  READ (1,1) (CCIN(1,K), K=1, MAXCON)
       .  .  .  DO (I=2,NM)
       .  .  .  .  DO (K=1, MAXCON) CCIN(I,K)=CCIN(1,K)
       .  .  .  ...FIN
       .  .  ...FIN
       .  .  IF (ECHO)
       .  .  .  WRITE(6,4) ENDTIM
       .  .  .  WRITE(6,3)
       .  .  .  DO (I=1,NM)
       .  .  .  .  WRITE(6,2) I,(CCIN(I,K),K=1,MAXCON)
       .  .  .  ...FIN
       .  .  ...FIN
 C     .  .
       .  .  WRITE(2) ENDTIM,NM,UDEPTH,((CCIN(I,K),K=1,MAXCON),I=1,NM)
       .  .  PRETIM = ENDTIM
       .  ...FIN
       ...FIN
       IF (PRETIM .LT. SIMLEN) CALL PUTERR(24, NUMERR, HLDERR)
 C
       REWIND 2
 C     RECORD THREE.....CHANNEL CROSS-SECTION DATA
 C          .....UWID -- WIDTH OF SEGMENT AT NODES
 C          .....UEL  -- ELEVATION OF NODE ABOVE BOTTOM
 C
 C***************************************************************
 C     CAUTION!
 C     IF CONCENTRATIONS ARE INPUT AT VARIOUS DEPTHS,
 C     THE CHANNEL CROSS-SECTIONAL DATA MUST BE INPUT
 C     AT THOSE SAME DEPTHS.
 C
 C***************************************************************
       READ(1,7) NWID, UWIDTH, DEL
       WHEN (NWID.EQ.0)
       .  E=0.
       .  DO (I=1,MXELEM)
       .  .  UWID(I)=UWIDTH
       .  .  UEL(I)=E
       .  .  E=E+DEL
       .  ...FIN
       ...FIN
```

```
          ELSE
          .  READ(1,1) (UWID(I),I=1,NWID)
          .  READ(1,1) ( UEL(I),I=1,NWID)
          .  IF(NWID.LT.MXELEM)
          .  .  DO (I = NWID + 1,MXELEM)
          .  .  .  UWID(I) = UWIDTH
          .  .  .  UEL(I) = UEL(I-1) + DEL
          .  .  ...FIN
          .  ...FIN
          ...FIN
          IF(ECHO)
          .  WRITE(6,8) (I,UWID(I),I=1,MXELEM)
          .  WRITE(6,9) (I, UEL(I),I=1,MXELEM)
          ...FIN
C
          RETURN
C
    1     FORMAT(8F10.0)
    2     FORMAT(2X,I3,1X,6(3X,1PE12.5),4X,1PE12.5)
    3     FORMAT(1H0,'ELEMENT',1X,
         1  3(3X,'CONC. OF',4X),3(1X,'CONC. ASSOC.',2X),2X,'CONTAMINANT'/
         2  9X,'SUSPENDED SAND',1X,'SUSPENDED SILT',1X,'SUSPENDED CLAY',
         3  3X,'WITH SAND',6X,'WITH SILT',6X,'WITH CLAY',4X,'DISSOLVED '
         4  'CONC.')
    4     FORMAT(1H0,8X,I10,'...DATA SET ENDING TIME')
    5     FORMAT(I10,I5,F10.0)
    6     FORMAT(1H0,48X,'UPSTREAM INFLOW WATER CONDITIONS')
  7       FORMAT (I5,2F10.0)
  8       FORMAT (1H0,58X,'X-SECTION WIDTHS'/4(27X,5(I3,1PE12.5)/))
  9       FORMAT (1H0,58X,'X-SECTION ELEVATIONS'/4(27X,5(I3,1PE12.5)/))
C
          END
```

(FLECS VERSION 22.46)

```
      -----------------------------------------
          SUBROUTINE WTRDAT(C, ECHO, NELEM)
C
C    THIS ROUTINE IS RESPONSIBLE FOR READING THE INITAL WATER CONDITIONS.
C
C      FORMAL PARAMETERS:
C        C     - INITIAL WATER CONDITIONS
C        ECHO  - LINE PRINT ECHO OPTION CONTROL VARIABLE (L*1)
C        NELEM - NUMBER OF VERTICAL ELEMENTS
C
C    CALLED BY: SERATRA
C
          INCLUDE 'ELMSIZ.PRM'
          LOGICAL*1 ECHO
            INTEGER SWITCH
C
          DIMENSION C(MXELEM,MAXCON)
C
C    ....INITAL WATER CONDITIONS
C
C      A SET OF INITIAL WATER CONDITIONS ARE READ FOR EACH NODE
C      OF THE SEGMENT (ELEMENTS ARE NUMBERED BEGINNING AT THE BOTTOM,
C      (ELEMENT 1), AND ENDING WITH THE SURFACE ELEMENT, (ELEMENT #NELEM)
C
C      IF THERE IS NO VERTICAL VARIATION, COLUMNS 1-5 CONTAIN A NEGATIVE
C      VALUE AND COLUMNS 6-15 CONTAIN THE CONSTANT VALUE. WHEN THE DATA
C      DOES VARY WITH DEPTH, A VALUE IS READ FOR EACH ELEMENT. THE UNITS
C      OF THE CONTAMINANT CONCENTRATIONS DEPEND UPON THE TYPE OF CONTAMIN
C      (RADIONUCLIDE..PC/KG OR PESTICIDE..KG/KG).
C
C                        THE UNITS OF THE CONTAMINANT CONCENTRATIONS DEPEND
C                        UPON THE TYPE OF CONTAMINANT (RADIONUCLIDE..PC/KG
C                        OR PESTICIDE..KG/KG). THE PARAMETERS ARE READ IN
C                        THE FOLOWING ORDER:
C
C          PARAMETER 1...CONCENTRATION OF SAND (KG/M**3)
C                    2...CONCENTRATION OF SILT (KG/M**3)
C                    3...CONCENTRATION OF CLAY (KG/M**3)
C                    4...CONCENTRATION OF CONTAMINANT ASSOCIATED WITH
C                          SAND
C                    5...CONCENTRATION OF CONTAMINANT ASSOCIATED WITH
C                          SILT
C                    6...CONCENTRATION OF CONTAMINANT ASSOCIATED WITH
C                          CLAY
C                    7...CONCENTRATION OF DISSOLVED CONTAMINANT
C
          DO (K=1,MAXCON)
          .  READ(1,2) SWITCH,VALUE
          .  WHEN(SWITCH .LT. 0)
C         .  .  *** PARAMETER DOES NOT VARY VERTICALLY ***
          .  .  DO (J=1,NELEM+1) C(J,K) = VALUE
          .  ...FIN
          .  ELSE
C         .  .  *** PARAMETER VARIES VERTICALLY ***
```

```
           .  .  READ(1,1) (C(J,K),J=1,NELEM+1)
           .  ...FIN
           ...FIN
           IF (ECHO)
           .  WRITE(6,3)
           .  WRITE(6,5)
           .  DO (J=1,NELEM+1)
           .  .  WRITE(6,4)J,(C(J,K),K=1,MAXCON)
           .  ...FIN
           ...FIN
C
           RETURN
C
1          FORMAT(8F10.0)
2          FORMAT(I5,F10.0)
3          FORMAT(1H0,52X,'INITIAL WATER CONDITIONS')
4          FORMAT(3X,I2,1X,6(3X,1PE12.5),4X,1PE12.5)
5          FORMAT(1H0,'ELEMENT',1X,
          1  3(3X,'CONC. OF',4X),3(1X,'CONC. ASSOC.',2X),2X,'CONTAMINANT'/
          2  9X,'SUSPENDED SAND',1X,'SUSPENDED SILT',1X,'SUSPENDED CLAY',
          3  3X,'WITH SAND',6X,'WITH SILT',6X,'WITH CLAY',4X,'DISSOLVED '
          4  'CONC.')
C
           END

(FLECS VERSION 22.46)

-----------------------------------
```


☆USGPO: 1982 — 559-092/0512