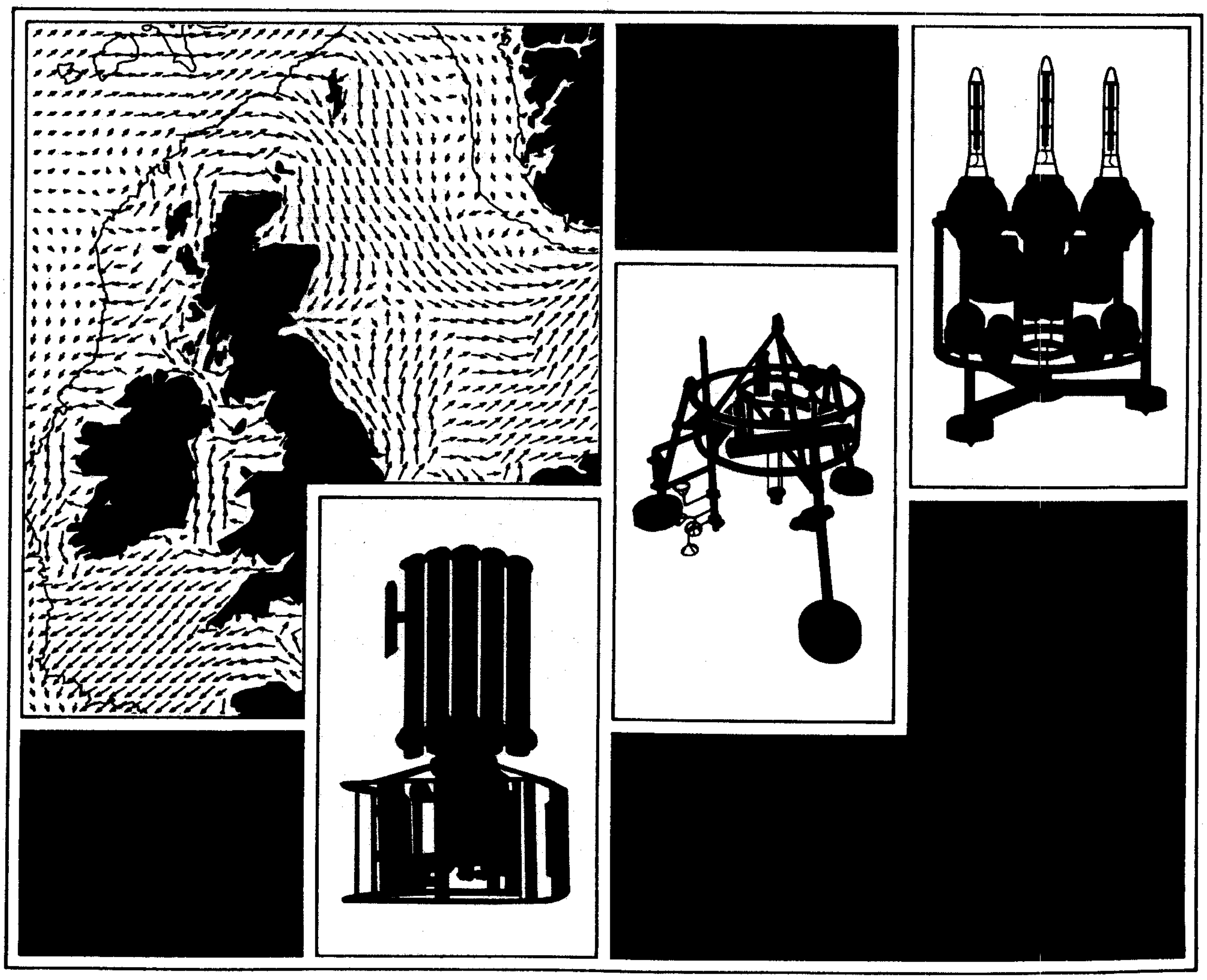
Proudman
Oceanographic
Laboratory

A unified framework for water quality modelling in shallow seas

J Wolf

Report No 19 1991



Natural Environment Research Council

PROUDMAN OCEANOGRAPHIC LABORATORY

Bidston Observatory,
Birkenhead, Merseyside, L43 7RA, U.K.

Telephone: 051 653 8633
Telex 628591 OCEANB G
Telefax 051 653 6269

Director: Dr. B.S. McCartney

Natural Environment Research Council

PROUDMAN OCEANOGRAPHIC LABORATORY

REPORT NO. 19

A unified framework for water quality modelling in shallow seas

J. Wolf

1991

structure does include the use of control variables which are read in to select various options, for example, the type of boundary condition. The horizontal grid uses spherical polar coordinates, allowing for the Earth's curvature. The model has been run for a number of years throughout the winter months so that an archive of wind-driven residuals is available.

4.2 'User-friendly' general purpose North Sea model

A depth-averaged model which has been widely disseminated among the North Sea Project community is described by Jones (1991). This provides the residual currents required to transport various substances around the North Sea. The aim is for the model to be available to workers in other fields, unfamiliar with modelling.

4.3 3-d turbulence energy model

Davies and Jones (1991) describe a model which solves the turbulence energy equation in order to provide the eddy viscosity for a 3-d current model. Grid boxes rather than the Galerkin method were found preferable in the vertical to resolve the structure of the boundary layer, although the turbulence energy model can be used to construct the optimal vertical modes for a Galerkin model (Davies, 1991), by giving representative eddy viscosity profiles.

4.4 Eddy-resolving frontal model

A density-evolving model including a single buoyancy equation rather than salt and heat separately has been developed by James (1986, 1987). The model uses the B-grid, with a sigma-coordinate transformation and grid-boxes in the vertical. It employs a time-split time integration scheme with a semi-implicit method for the barotropic mode. An application to the Flamborough Head front in the North Sea is given in James (1989) which examines the effect of increasing the horizontal resolution of the model. A sponge open boundary condition was found to be the most effective at avoiding spurious reflections.

4.5 Transport model

Hainbucher et al (1987) looked at the effect of introducing conservative passive tracers using a Lagrangian particle-tracking technique. The flow field was derived from a three-dimensional model of the North Sea described by Backhaus (1985), using fixed levels in the vertical (12 in summer and 7 in winter). The bottom level varies in thickness to accommodate changes in topography. In shallow water the levels drop out until the model is equivalent to a depth-averaged model with a single layer in the vertical. A semi-implicit time-integration scheme is used, to extend the time-step. This gives poor representation of the tidal phase but has little effect on the calculation of residual currents. Various tracers can be simulated including fish larvae (Backhaus, 1990).

4.6 Sediment transport model

Puls (1987) and Puls and Sundermann (1990) describe a model also based on the Backhaus (1985) 3-dimensional current model with a Lagrangian particle-tracking model for suspended sediment superimposed. Mud (cohesive) and non-cohesive sediment is considered and 12 layers in the bottom sediment are introduced. The problems in sediment transport modelling are still in the physics rather than in the numerical methods for solution. Rates of erosion, particularly of cohesive sediment are difficult to define for all conditions. Sheng (1986) and Sheng et al (1990) also discuss this subject. There is still insufficient field data to resolve some of the problems. The modelling and measurement work must go hand in hand (Krohn et al, 1991).

DOCUMENT DATA SHEET

AUTHOR WOLF, J	PUBLICATION DATE 1991
TITLE A unified framework for water quality modelling in shallow seas.	
REFERENCE Proudman Oceanographic Laboratory, Report No. 19, 46pp.	

ABSTRACT

A unified framework for numerical sea modelling is presented, which shows the linkages between various physical processes and suggests some mechanisms for incorporating them in a computer program. Until recently the development of numerical models for oceanography has generally been carried out independently by individual research workers. However, with the computer capacity available and the potential for highly complex model systems, there is now a need for more collaboration between individuals and groups of researchers. Different processes or aspects of the model can be developed in isolation and then fitted back into the framework. This allows ultimately more freedom and flexibility to individuals to investigate details of processes or numerical techniques and at the same time a state-of-the-art predictive model can be constructed. The particular application here is shallow sea water quality modelling, but the approach can be applied to any complex model system.

This work was funded by MAST contract MAST-0050-C.

ISSUING ORGANISATION Proudman Oceanographic Laboratory Bidston Observatory Birkenhead, Merseyside L43 7RA UK Director: Dr B S McCartney	TELEPHONE 051 653 8633 TELEX 628591 OCEAN BG TELEFAX 051 653 6269

KEYWORDS NUMERICAL MODELLING WATER QUALITY SEDIMENT TRANSPORT STORM SURGE PREDICTION	SHALLOW WATER NORTH SEA TIDAL MODELS	CONTRACT PROJECT MNS-325 PRICE £15

Copies of this report are available from:
The Library, Proudman Oceanographic Laboratory.

REFERENCES

ARAKAWA, A. & LAMB, V.R. 1977 Computational design of the basic dynamical processes of the UCLA general circulation model. pp. 174-265 in, General circulation models of the atmosphere, (ed. J. Chang). New York: Academic Press. 337pp. (Methods in Computational Physics, 17).

BACKHAUS, J.O. 1985 A three-dimensional model for the simulation of shelf sea dynamics. Deutsche Hydrographische Zeitschrift, 38(4), 165-187.

BACKHAUS, J.O. 1990 On the atmospherically induced variability of the circulation of the Northwest European shelf sea and related phenomena - a model experiment. pp.93-134 in Modeling marine systems, Volume 1, (ed. A.M. Davies). Boca Raton, Florida: CRC Press. 297pp.

BATCHELOR, G.K. 1967 An introduction to fluid mechanics. Cambridge: University Press. 615pp.

BLUMBERG, A.F. & HERRING, H.J. 1987 Circulation modelling using orthogonal curvilinear coordinates. pp. 55-88 in, Three-dimensional models of marine and estuarine dynamics (ed. J.C.J. Nihoul and B.M. Jamart). Amsterdam: Elsevier. 629pp. (Elsevier Oceanography Series, 45).

DAVIES, A.M. 1991 On the accuracy of finite difference and modal methods for computing tidal and wind wave current profiles. International Journal for Numerical Methods in Fluids, 12, 101-124.

DAVIES, A.M. & JONES, J.E. 1991 On the numerical solution of the turbulence energy equations for wave and tidal flows. International Journal for Numerical methods in Fluids, 12, 17-41.

DAVIES, A.M. & STEPHENS, C.V. 1983 Comparison of the finite difference and Galerkin methods as applied to the solution of the hydrodynamic equations. Applied Mathematical Modelling, 7, 226-240.

FLATHER, R.A., PROCTOR, R. AND WOLF, J. 1991 Oceanographic forecast models. pp.15-30 in, Computer modelling in the environmental sciences, (ed. D.G. Farmer & M.J. Rycroft). Oxford University Press. 379pp.

GESAMP Working Group 25 on water quality modelling (chairman J. Blanton) 1991. (Unpublished manuscript).

GIBSON, J.K. 1982 The DOCTOR system - a DOCumenTary Oriented programming system. European Centre for Medium Range Weather Forecasts, Technical Memorandum No. 52, 17pp.

HAINBUCHER, D., POHLMANN, T. & BACKHAUS, J. 1987 Transport of conservative passive tracers in the North Sea: first results of a circulation and transport model. Continental Shelf Research, 7(10), 1161-1179.

HOWARTH, M.J. 1988 The (NERC) North Sea Community Project, 1987-1992. pp. 247-260 in, Advances in underwater technology, ocean science and offshore engineering. Vol. 16. Oceanology '88. London: Graham & Trotman for the Society for Underwater Technology. 304pp.

HUTHNANCE, J.M. 1991 Community research projects - the North Sea Project: a progress report. Ocean Challenge, 1 (Spring), 31-33.

CONTENTS Page no.

1. INTRODUCTION 7

2. SHALLOW SEA WATER QUALITY MODELLING 7

2.1 Equations 8
2.2 Physical processes 8
2.3 A model framework 8

3. NUMERICAL METHODS 9

3.1 Grid selection 9
3.2 Time integration method 10
3.3 Advection-dispersion 11
3.4 Boundary conditions 11

4. EXISTING MODELS 11

4.1 Tide-surge 2-d model 11
4.2 'User-friendly' general purpose North Sea model 12
4.3 3-d turbulence energy model 12
4.4 Eddy-resolving frontal model 12
4.5 Transport model 12
4.6 Sediment transport model 12
4.7 Phytoplankton model 13

5. CONCLUSIONS 13

Acknowledgements 13

References 14

Figure 1 *Conceptual model for a water quality model, from GESAMP Working Group 25.* 17

Figure 2 *Nearshore fine-resolution model framework.* 18

Figure 3 *Detailed subroutine structure.* 19

Figure 4 *Arakawa A-, B- and C- grids.* 20

APPENDIX Sample FORTRAN code 21

ROED, L.P. & COOPER, C.K. 1986 Open boundary conditions in numerical ocean models. pp.411-436 in, Advanced physical oceanographic numerical modelling, (ed.J.J.O'Brien). Dordrecht: D. Reidel. 608pp. (NATO ASI Series C. Mathematical and physical sciences, vol.186).

SHENG, Y.P. 1986 Modeling bottom boundary layer and cohesive sediment dynamics in estuarine and coastal waters. pp.360-400 in, Estuarine cohesive sediment dynamics, (ed.A.J.Mehta). Berlin: Springer-Verlag. 473pp. (Lecture Notes on Coastal and Estuarine Studies No.14).

SHENG, Y.P., COOK, V., PEENE, S., ELIASON, D., SCOFIELD, S., AHN, K-M. & WANG, P.F. 1990 A field and modeling study of fine sediment transport in shallow waters. pp.113-122 in, Estuarine and coastal modeling, (ed. M.L. Spaulding). New York: American Society of Civil Engineers. 536pp.

SPAULDING, M.L. 1984 A vertically averaged circulation model using boundary-fitted coordinates. Journal of Physical Oceanography, 14 (5), 973-982.

TETT, P.B. 1990 A three layer vertical and microbiological processes model for shelf seas. Proudman Oceanographic Laboratory, Report, 14, 85pp.

THACKER, W.C. 1977 Irregular grid finite-difference techniques: simulations of oscillations in shallow circular basins. Journal of Physical Oceanography, 7(2), 284-292.

WAMDI Group 1988 The WAM model, a third generation ocean wave prediction model. Journal of Physical Oceanography, 18(12), 1775-1810.

WOLF, J. 1983 A comparison of a semi-implicit with an explicit scheme in a three-dimensional hydrodynamic model. Continental Shelf Research, 2(4), 215-229.

YANG, G., BELLEUDY, P. & TEMPERVILLE, A. 1991 A higher-order Eulerian scheme for coupled advection-diffusion transport. International Journal for Numerical Methods in Fluids, 12(1), 43-58.

1. INTRODUCTION

Until recently the development of numerical models for oceanography has generally been carried out independently by individual research workers. However, with the computer capacity available and the potential for highly complex model systems, there is now a need for more collaboration between individuals and groups of researchers. Interactive models in which different processes or aspects of the model can be developed in isolation and then fitted back into a unifying framework are required. This allows ultimately more freedom and flexibility to individuals to investigate details of processes or numerical techniques and at the same time a state-of-the-art predictive model can be constructed. The idea of teamwork and standardisation of model code has been used for some years in atmospheric modelling (Gibson, 1982) and more recently in wave modelling (the WAMDI group, 1988). The demand for a water quality model of the North Sea as a management tool has focussed the need for a more structured approach to shallow sea modelling (Howarth, 1988; Prandle, 1990). This is a more specific description of what such a model must contain than previously attempted (e.g. GESAMP Working Group 25, 1991). It aims to examine some of the practical problems to be solved in constructing such a model.

A flowchart of the physical processes which need to be included is given in §2, with some discussion of the philosophy behind it. In §3 some of the issues which must be addressed are discussed briefly, for example, what grid scheme should be used and how flexible its selection should be. Many of these questions do not have a single solution, however one of the benefits of providing a framework is to highlight the problem areas and focus discussion. Also the list of possible options may be illuminating to workers in different fields. The discussion is mainly on the hydrodynamics options; sediment, chemical and biological modelling being included in outline only. Some existing models and their possible place within the framework is given in §4. A sample FORTRAN program to illustrate the program structure is given in the appendix.

2. SHALLOW SEA WATER QUALITY MODELLING

The main hydrodynamic processes are tides and wind-driven flow which affect the residual transport of momentum, heat, salt, suspended sediment and contaminants. The latter include chemical and biological substances, which may interact in complex ways (also involving suspended sediment). The modelling requires solution of partial differential equations expressing conservation of the various substances. One of the main problems arises from the fact that, to keep the computational effort within bounds, the equations are generally averaged over different time-scales, representing the most important processes directly, but introducing parametrisation of the higher frequency effects (see e.g. Batchelor, 1967; Pedlosky, 1979). Most of the complexity of the solution procedure arises from trying to introduce more sophisticated schemes to calculate the parameters in as physically realistic a way as possible. Also there are many time-scales involved from (typically) hourly for tide and wind effects to the monthly time-scale required to represent the seasonal cycle due to solar heating (Howarth, 1988). Another problem area is the selection of suitable numerical techniques which correctly reproduce the physics. This is a large area of research in its own right, which has vital implications in practice.

2.1 Equations

The first requirement is to define the problem to be solved in terms of the fundamental equations and any simplifications and assumptions which are to be introduced. From the hydrodynamics point of view the main equations are: the continuity equation (conservation of mass), the equation of motion in a rotating frame (conservation of momentum) and the transport equations for various substances (individual conservation equations). The derivation of the equations is given elsewhere e.g. Batchelor

NEARSHORE FINE-RESOLUTION MODEL FRAMEWORK

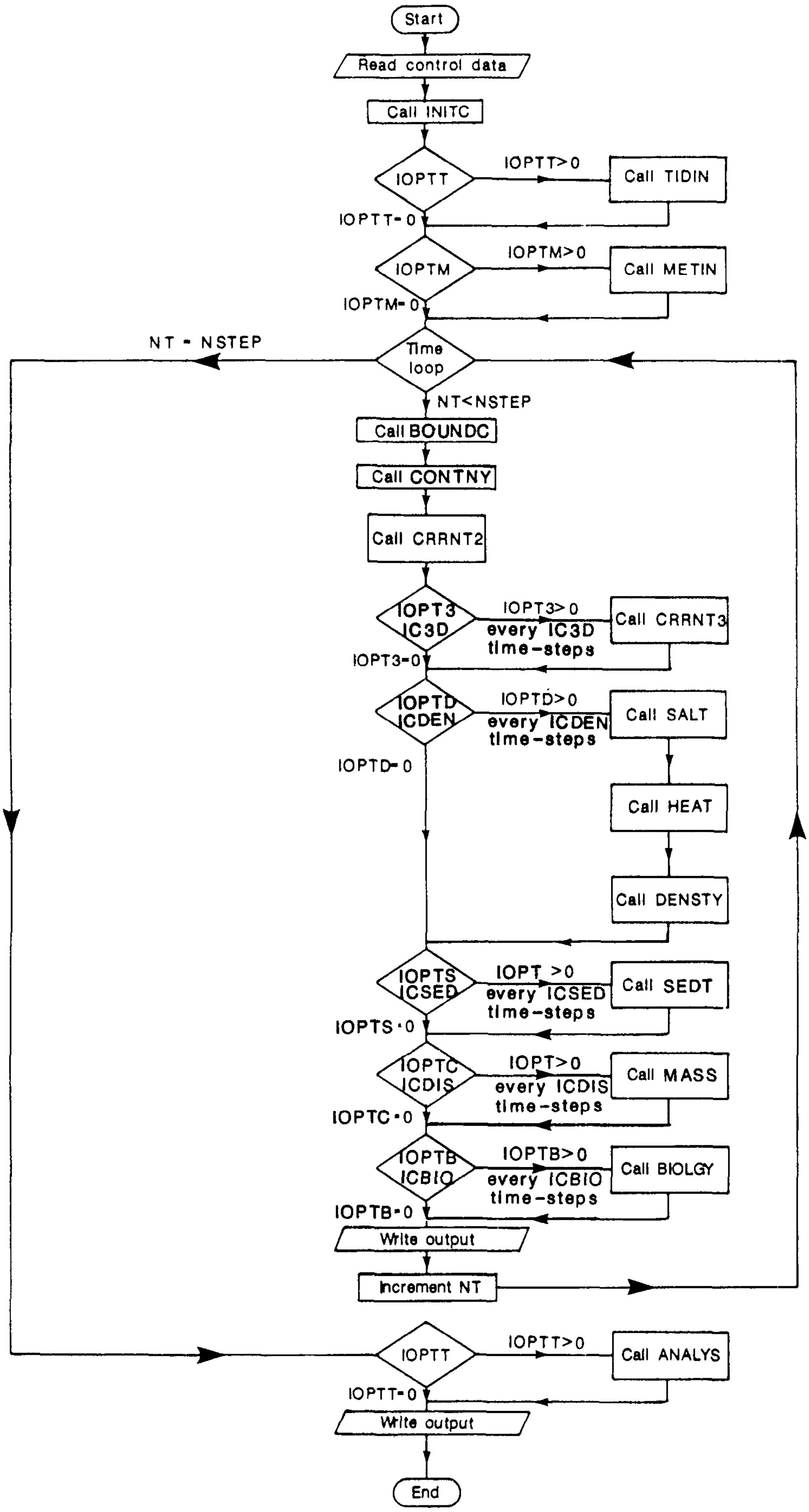


Figure 2 *Nearshore fine-resolution model framework.*

Fig. 3 identifies more of the detail which may be required to actually solve the equations for the different processes, although all possible options are not explicitly included. The subroutine INITC includes the setting up of the model grid, which may include coordinate transformations. These are discussed in §3.1. An important feature, which may readily be seen in the flowchart, is that many subroutines call the subroutine ADVDIS whose function is to solve the advection-dispersion equation at a particular time step. This is fundamental to the transport of all substances including momentum, but there is no optimum numerical method of solving this, as will be discussed in §3.3. However the framework allows this to be identified as a central problem to be tackled before any model is constructed. Another important topic is the specification of eddy viscosity for the calculation of three-dimensional currents (also the related problem of eddy diffusivities). This arises from the time-averaging of the nonlinear advective terms in the fundamental equations. Some possible options are included e.g an algebraic formulation of viscosity, or solution of the turbulence energy equation plus possibly further equations in turbulent quantities.

3. NUMERICAL METHODS

Two main methods are commonly used: the finite difference method and the finite element method. Both involve splitting up the sea area by means of a grid, at discrete points of which the solution is obtained. The finite difference method splits up the area into regular grid-boxes and assumes the solution to be constant in each grid-box. Partial differentials are replaced by differences between the dependent variables at adjacent grid-boxes. The finite element method splits up the solution into the sum of a weighted set of basis functions. The solution is found by a variational method. The latter is ideally suited to elliptic boundary-value problems although it has been adapted to time-evolving problems (Le Provost, 1986). A most useful feature of the finite element method is that it can use variable size and shape of elements which allows improved resolution of boundaries and other areas of interest. A combination of finite differences in the horizontal and finite elements in the vertical has been found useful (Davies and Stephens, 1983; Davies, 1991) but probably the most flexible method is the finite difference method throughout, which is chosen here. The selection of this option narrows down the field of decisions which must be made which is advantageous, while losing some of the potential advantages of smoothness and accuracy of the solution for a given resolution and amount of computational effort. However the possibility of including a grid transformation option in the equations allows some flexibility in tailoring the grid to fit boundaries more accurately.

3.1 Grid selection

Three main types of finite difference grid in the horizontal have been identified (Arakawa and Lamb, 1977), usually termed the A-, B- and C-grids, which include various degrees of staggering of the dependent variables between adjacent grid boxes (giving centred differences and higher accuracy in discretisation of the partial differentials for the same grid resolution). The grid layouts are shown in Fig.4. In particular the B- and C-grids have been found to be useful in hydrodynamic modelling for different applications. The C-grid in which the elevation and two components of velocity are all at different points in the grid is ideally suited to modelling wave propagation. However when the advective terms become important the B-grid becomes preferable, since the averaging required in velocities on the C-grid degrades the solution. It may be that both grids will be required in different parts of the model, so that interpolation routines must be included, however this would be envisaged as being an interface at the beginning or end of a run, not to be performed at short time intervals since this would degrade the solution.

As already indicated some transformation of the equations can be performed to allow variation of the grid resolution and better boundary fitting. This is particularly useful in the vertical, for example when the so-called sigma-coordinate system is introduced (e.g. James, 1987). This maps the total water

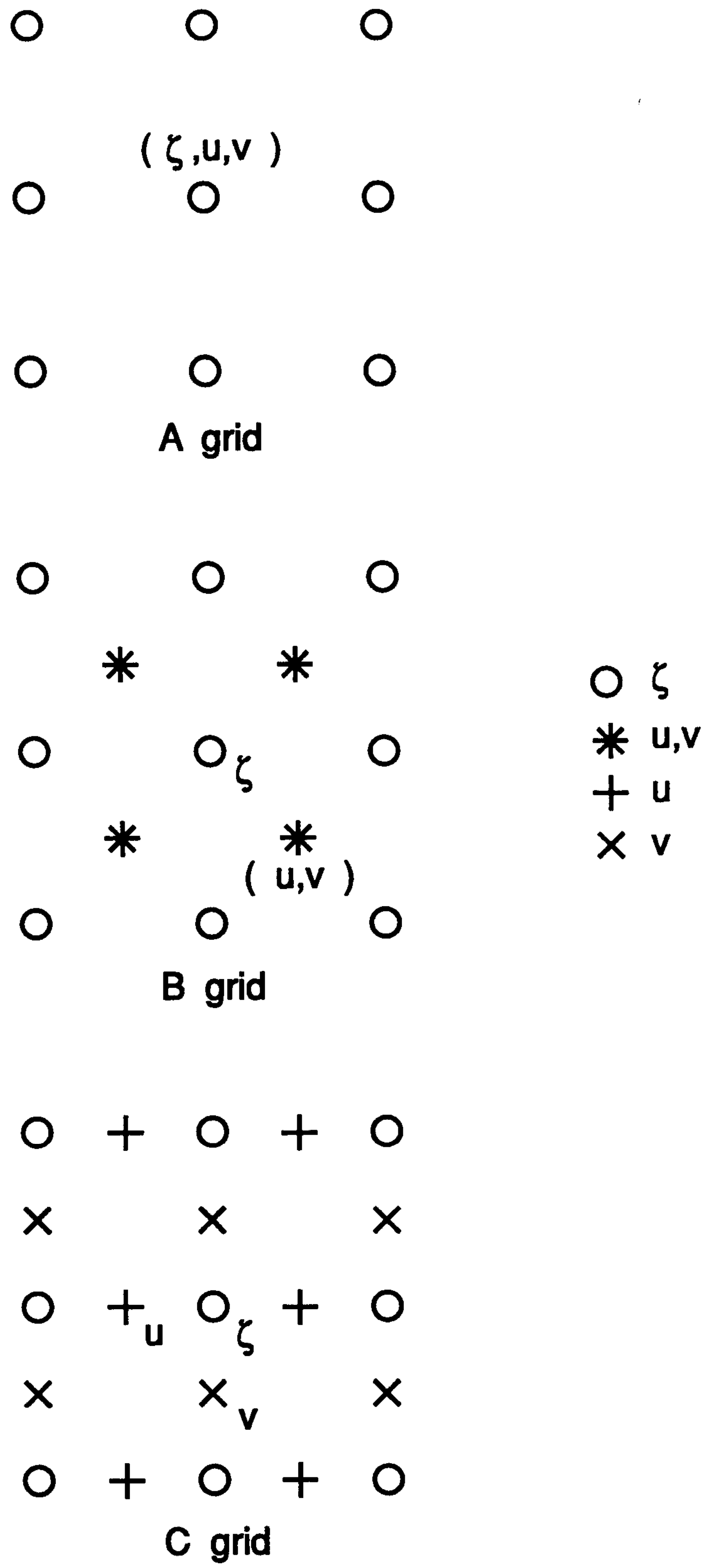


Figure 4 *Arakawa A-, B- and C- grids.*

3.3 Advection-dispersion

As indicated in §2, the advection-dispersion equation is central to the transport of all substances. The dispersion is mainly due to turbulent diffusion and can be parametrised at various levels in a similar way to eddy viscosity. One method of treating advection-dispersion is the Lagrangian particle-tracking technique which relies on computing the statistics of a sufficiently large number of particles which are introduced into the flow field. Turbulence may be introduced by various means, e.g. the Monte-Carlo random walk method. The particle-tracking method avoids some of the problems of the finite-difference method although the latter is consistent with the rest of the model structure. The problems are inherent in the finite difference approach, in particular how to treat a discontinuity in some quantity. Finite differences rely on the continuity of the solution and sharp gradients are difficult to handle. The two extremes of the finite difference method for treating the horizontal gradient term are the 'centred' difference method which tends to produce spurious oscillations and the 'upwind' difference method which avoids oscillations but introduces numerical diffusion i.e. smooths the gradient. Various hybrid and modified schemes have been developed for this problem e.g. James (1986). Schemes which use two or more time levels and several grid-points in space can be developed (Noye and Tan, 1989; Noye, 1991, Yang et al, 1991), which tend to deal well with one specific application but can be expensive to run. The inclusion of various options possibly including the particle-tracking approach may be desirable.

3.4 Boundary conditions

There are generally two types of boundary to be considered: the closed (land) boundary and the open (sea) boundary. The former is usually treated by applying a condition of no flow across the boundary. The question of the correct position and orientation of the boundary must be considered and this may be partly dealt with by grid transformation as in §3.1. The boundary may be regarded as mobile in which case the position of the boundary must be computed at each time step.

Open boundaries may be regarded as totally artificial, required because of the finite amount of computation which can be achieved particularly at very high resolution. The aim must be to make the boundary as transparent as possible. External forcing must be applied, but outgoing energy must be transmitted without reflection. Given that there are constraints on computation, the ideal boundary condition for all situations is impossible to define. Various options may be required and Roed and Cooper (1986) review many of these.

4. EXISTING MODELS

Example models are now described to illustrate some of the problems and some of the solutions which exist at present. The particular examples which are chosen are by no means an exhaustive and unbiassed sample. They represent those most familiar to the author and are generally those involved in the North Sea Project and the MAST program MAST-0050-C, projects which have particularly stimulated the interest in interdisciplinary modelling and the need to have a common framework for communication between modellers.

4.1 Tide-surge 2-d model

Surge and tide forecasting has been carried out at the UK Meteorological Office since 1978 using a depth-averaged model covering the UK continental shelf. New developments include finer resolution grids and the effects of surface waves (Flather et al, 1991). The model uses an explicit time-integration scheme. Optimisation has been in terms of reducing computer storage and time for a production model on specific computers rather than developing a totally flexible model. However the

```
C
C     COMMON BLOCKS AND SUBROUTINES ARE LISTED ALPHABETICALLY.
C     LABELS AND VARIABLE NAMES ARE INTENDED TO BE AS SELF-EXPLANATORY
C     AS POSSIBLE.
C     ALL SUBROUTINES MUST CONTAIN COMMENTS ON THEIR PURPOSE, AUTHOR
C     AND LAST DATE OF UPDATE, THE NAME OF THE CALLING PROGRAM AND ANY
C     SUBROUTINES ACCESSED (EXTERNALS), PLUS DEFINITIONS OF ALL LOCAL
C     VARIABLES.
C
C     LIST OF EXTERNALS
C     -----------------
C
C     NAME                         DESCRIPTION
C     ----                         -----------
C
C    *ADVDIS*      SOLVES ADVECTION-DISPERSION EQUATION
C
C    *ANALYS*      TIDAL ANALYSIS
C
C    *BIOLGY*      SOLVE PHYTOPLANKTON DYNAMICS
C
C    *BOUNDC*      TREAT BOUNDARY CONDITIONS
C
C    *BSTRES*      BOTTOM STRESS CALCULATIONS
C
C    *CONTNY*      SOLVE CONTINUITY EQUATION FOR SURFACE ELEVATIONS
C
C    *CRRNT2*      SOLVE 2-D MOMENTUM EQUATIONS
C
C    *CRRNT3*      SOLVE 3-D MOMENTUM EQUATIONS
C
C    *DENSTY*      CALCULATE DENSITY
C
C    *DISSIP*      CALCULATE TURBULENCE DISSIPATION RATE
C
C    *HEAT*        SOLVE HEAT BUDGET
C
C    *HEDDY*       SET UP HORIZONTAL EDDY VISCOSITY, DIFFUSIVITY
C
C    *INITC*       SET UP INITIAL CONDITIONS
C
C    *INCDAT*      INCREMENT DATE
C
C    *INTEGR*      TIME-AVERAGING OF VARIABLES
C
C    *MASS*        MASS TRANSPORT OF CONTAMINANT
C
C    *METIN*       MET FORCING
C
C    *OUTPA*       OUTPUT ARRAYS
C
C    *OUTPT*       OUTPUT TIME SERIES
C
C    *PRINT*       OUTPUT REPORT TO UNIT 6
C
C    *SALT*        SOLVE SALINITY CONSERVATION EQUATION
C
C    *SEDT*        SOLVE SEDIMENT TRANSPORT EQUATIONS (VARIOUS FRACTIONS)
C
C    *TIDIN*       SET UP TIDAL BOUNDARY CONDITIONS
C
C    *TLENG*       CALCULATE TURBULENCE LENGTH SCALE
C
C    *TRANSH*      TRANSFORM HORIZONTAL COORDINATES (OPTIONAL)
C
C    *TRANSV*      TRANSFORM VERTICAL COORDINATE (OPTIONAL)
C
C    *TURBEN*      SOLVE TURBULENCE ENERGY EQUATIONS (VARIOUS CLOSURES)
C
C    *VEDDY*       SET UP VERTICAL EDDY VISCOSITY, DIFFUSIVITY
C
C    *WAVIN*       INCLUDE WAVE EFFECTS
C
C**
C***********************************************************************
C
C     START OF MAIN PROGRAM
C
C***********************************************************************
C     include 'param.inc' *** used in all SUBROUTINES ***
C
      PARAMETER( NC = 50, NR = 100, NZ = 10, NL = 2 ,
     1           NOBZ = 200, NOBU = 200, NOBV = 200 , NCON = 2 )
C
C*    NAME       TYPE       PURPOSE
C     ----       ----       -------
C    *NC*        INTEGER    NO. OF COLUMNS IN GRID (WEST-EAST)
C    *NR*        INTEGER    NO. OF ROWS IN GRID    (NORTH-SOUTH)
```

4.7 Phytoplankton model

Tett (1990) has produced an integrated physical and biological model which employs three layers (including a bottom sediment layer) in the vertical. There is no horizontal dimension. The model employs a 1-day time step and describes the change in depth of the thermocline and the growth and decay of phytoplankton depending on the supply of light, oxygen and nutrients. As with sediment transport, (probably to an even larger degree since the model requires knowledge of the suspended sediments), there are still many unknowns in the mechanisms which make a very complex numerical solution seem pointless. The model has been constructed to give physically sensible results and illustrates how a full understanding of the subject is required to produce a model which is self-consistent.

5. CONCLUSIONS

All the above models have a great deal in common in that they are forward-time-stepping finite difference models. Although they treat the grid-discretisation in different ways and solve equations for different variables they are therefore not totally incompatible. The great advantage to be gained in relating them to a common framework is that differences between models can be examined without introducing too many variables. The effect of one time-integration scheme or another may be compared without having to account for different horizontal grid schemes for example. Also the benefits of work in one field, e.g. on the advection-diffusion problem, can be readily accessed by others who may be more interested in the elucidation of the resuspension of cohesive sediments. There is a tendency for specialists in one area to be unwilling to become too involved in another speciality and to reject the inclusion of, say, state-of-the-art physics in a biological model because the development of the equations for the biological processes is not complete. However there is an argument for including the best possible physics so that the errors in the physics cannot contaminate the processes which are being tested. The final goal of a fully interdisciplinary model may yet be some distance away, but it must surely be worth striving for.

Acknowledgements Thanks are due to John Huthnance and David Prandle at Proudman Oceanographic Laboratory for encouragement and to Eric Deleersnijder of MUMM for much useful criticism.

```
C     ----         ----        -------
C     *IC3D*       INTEGER     COUNTER FOR CALCULATION OF 3D CURRENTS (DEL3/DELT)
C     *ICBIO*      INTEGER     COUNTER FOR INCLUSION OF BIOLOGY (DELP/DELT)
C     *ICDEN*      INTEGER     COUNTER FOR UPDATE OF DENSITY FIELD (DELD/DELT)
C     *ICDIS*      INTEGER     COUNTER FOR ADVECTION/DISPERSION (DELC/DELT)
C     *ICMET*      INTEGER     COUNTER FOR MET INPUT (DELM/DELT)
C     *ICOUT*      INTEGER     COUNTER FOR OUTPUT OF ARRAYS, SPOT VALUES (DELO/DELT)
C     *ICSED*      INTEGER     COUNTER FOR INCLUSION OF SEDIMENT TRANSPORT (DELS/DELT)
C     *ICWAV*      INTEGER     COUNTER FOR INCLUSION OF WAVES (DELW/DELT)
C
C-----------------------------------------------------------------------
C      include 'crrnts.inc' *** as used in SUBROUTINES ADVDIS, ANALYS, BIOLGY,
C                           BOUNDC, BSTRESS, CONTNY, CRRNT2, CRRNT3, DISSIP, HEAT,
C                           HEDDY, INITC, INTEGR, MASS, OUTPA, OUTPT, SALT, SEDT,
C                           TLENG, TURBEN, VEDDY, WAVIN ***
C
C*     COMMON *CRRNTS* - 2-D (DEPTH-AVERAGED) AND 3-D (DEVIATIONS) CURRENTS
C
       COMMON/CRRNTS/ U1(NC,NR,NZ), U2(NC,NR,NZ), V1(NC,NR,NZ),
     1                V2(NC,NR,NZ), W1(NC,NR,NZ), W2(NC,NR,NZ),
     2                UD1(NC,NR), UD2(NC,NR), VD1(NC,NR), VD2(NC,NR)
C
C-----------------------------------------------------------------------
C
C*     NAME         TYPE        PURPOSE
C      ----         ----        -------
C     *UD1*        REAL        EAST COMPONENT OF 2-D CURRENT AT LOWER TIME LEVEL
C     *U1*         REAL        EAST COMPONENT OF 3-D CURRENT AT LOWER TIME LEVEL
C     *UD2*        REAL        EAST COMPONENT OF 2-D CURRENT AT HIGHER TIME LEVEL
C     *U2*         REAL        EAST COMPONENT OF 3-D CURRENT AT HIGHER TIME LEVEL
C     *VD1*        REAL        NORTH COMPONENT OF 2-D CURRENT AT LOWER TIME LEVEL
C     *V1*         REAL        NORTH COMPONENT OF 3-D CURRENT AT LOWER TIME LEVEL
C     *VD2*        REAL        NORTH COMPONENT OF 2-D CURRENT AT HIGHER TIME LEVEL
C     *V2*         REAL        NORTH COMPONENT OF 3-D CURRENT AT HIGHER TIME LEVEL
C     *W1*         REAL        VERTICAL COMPONENT OF CURRENT AT LOWER TIME LEVEL
C     *W2*         REAL        VERTICAL COMPONENT OF CURRENT AT HIGHER TIME LEVEL
C
C-----------------------------------------------------------------------
C      include 'depths.inc' *** as used in SUBROUTINES ADVDIS, BIOLGY, BOUNDC,
C                           BSTRESS, CONTNY, CRRNT2, CRRNT3, DISSIP, HEAT, HEDDY,
C                           INITC, INTEGR, MASS, METIN, OUTPA, OUTPT, PRINT, SALT,
C                           SEDT, TLENG, TRANSH, TRANSV, TURBEN, VEDDY, WAVIN ***
C
C*     COMMON *DEPTHS* - TOPOGRAPHIC DATA
C
       COMMON/DEPTHS/ DEP(NC,NR), DEPMIN
C
C-----------------------------------------------------------------------
C
C*     NAME         TYPE        PURPOSE
C      ----         ----        -------
C     *DEP*        REAL        2-D ARRAY OF MEAN WATER DEPTHS (METRES)
C     *DEPMIN*     REAL        MINIMUM DEPTH CONSTANT
C
C-----------------------------------------------------------------------
C      include 'elev.inc' *** as used in SUBROUTINES ADVDIS, ANALYS, BIOLGY,
C                         BOUNDC, BSTRESS, CONTNY, CRRNT2, CRRNT3, DISSIP, HEAT,
C                         HEDDY, INITC, INTEGR, MASS, OUTPA, OUTPT, SALT, SEDT,
C                         TLENG, TURBEN, VEDDY, WAVIN ***
C
C*     COMMON *ELEV* - SURFACE ELEVATIONS (RELATIVE TO UNDISTURBED M.W.L.)
C
       COMMON/ELEV/ ZETA1(NC,NR), ZETA2(NC,NR)
C
C-----------------------------------------------------------------------
C
C*     NAME         TYPE        PURPOSE
C      ----         ----        -------
C     *ZETA1*      REAL        2-D ARRAY OF SURFACE ELEVATION AT LOWER TIME LEVEL
C     *ZETA2*      REAL        2-D ARRAY OF SURFACE ELEVATION AT HIGHER TIME LEVEL
C
C-----------------------------------------------------------------------
C      include 'grid.inc' *** as used in SUBROUTINES ADVDIS, BIOLGY, BOUNDC,
C                         BSTRESS, CONTNY, CRRNT2, CRRNT3, DISSIP, HEAT, HEDDY,
C                         INITC, INTEGR, MASS, PRINT, SALT, SEDT, TLENG, TRANSH,
C                         TRANSV, TURBEN, VEDDY, WAVIN ***
C
C*     COMMON *GRID* - MODEL GRID LAYOUT DATA
C
       COMMON/GRID/ GX1(NC,NR), GY1(NC,NR), GZ1(NC,NR,NZ), GX2(NC,NR),
     1              GY2(NC,NR), GZ2(NC,NR), IN1(NC,NR), IN2(NC,NR),
     2              IN3(NC,NR), IN4(NC,NR), DELX, DELY, DELZ, IGRID
C
C-----------------------------------------------------------------------
C
C*     NAME         TYPE        PURPOSE
C      ----         ----        -------
C     *GX1*        REAL        STORAGE OF TRANSFORM DATA E.G. DERIVATIVES
```

JAMES, I.D. 1986 A front-resolving sigma coordinate sea model with a simple hybrid advection scheme. Applied Mathematical Modelling, 10(2), 87-92.

JAMES, I.D. 1987 A general three-dimensional eddy-resolving model for stratified seas. pp.591-608 in, Three-dimensional models of marine and estuarine dynamics, (ed.J.C.J.Nihoul & B.M.Jamart). Amsterdam: Elsevier. 629pp. (Elsevier Oceanography Series, 45).

JAMES, I.D. 1989 A three-dimensional model of circulation in a frontal region of the North Sea. Deutsche Hydrographische Zeitschrift, 42(3-6), 231-247.

JAMES, I.D. 1990 Numerical modelling of density-driven circulation in shelf seas. pp.345-372 in, Modeling Marine Systems, Volume 2, (ed. A.M. Davies), Boca Raton, Florida: CRC Press. 442pp.

JONES, J.E. 1991 A general purpose model for the North Sea Project. NERC News, 16, 25-26.

KROHN, J., MULLER, A. & PULS, W. 1991 Pollutant transport monitoring and prediction by mathematical modelling: North Sea and adjacent estuaries. Marine Pollution Bulletin, 23, 699-702.

LE PROVOST, C. 1986 On the use of finite element methods for ocean modelling. pp.557-580 in, Advanced physical oceanographic numerical modelling, (ed.J.J.O'Brien). Dordrecht: D.Reidel. 608pp. (NATO ASI Series C. Mathematical and physical sciences, vol.186).

MARCHUK, G.I. 1975 Methods of numerical mathematics. New York: Springer-Verlag.

NOYE, J. 1987 Time-splitting the one-dimensional transport equation. pp. 271-295 in, Numerical modelling: applications to marine Systems, (ed. J.Noye). Amsterdam: North-Holland. 295pp.

NOYE, B.J. & TAN, H.H. 1989 Finite difference methods for solving the two-dimensional advection-diffusion equation. International Journal for Numerical Methods in Fluids, 9(2), 75-98.

NOYE, B.J. 1991 Some three-level finite difference methods for simulating advection in fluids. Computers and Fluids, 19(1), 119-140.

O'BRIEN, J.J. 1986 Time integration schemes. pp.155-163 in, Advanced physical oceanographic numerical modelling, (ed.J.J.O'Brien). Dordrecht: D.Reidel. 608pp. (NATO ASI Series C. Mathematical and physical sciences, vol.186).

PEDLOSKY, J. 1979 Geophysical fluid dynamics. New York: Springer Verlag, 624pp.

PRANDLE, D. 1991 A 5-year scientific research programme for managing coastal seas. Marine Pollution Bulletin, 23, 63-68.

PULS, W. 1987 Simulation of suspended sediment dispersion in the North Sea. International Council for the Exploration of the Sea, Council Meeting, Hydrography Committee, C.M.1987/C:37, Ref.MEQC, 24pp.

PULS, W. & SUNDERMANN, J. 1990 Simulation of suspended sediment dispersion in the North Sea. pp.356-372 in, Residual currents and long-term transport, (ed.R.T.Cheng). New York: Springer Verlag. 544pp. (Coastal and Estuarine Studies, 38)

```
C     *INPTD*     INTEGER   SWITCH TO SELECT DENSITY FIELD INITIAL ARRAY (0,1)
C     *INPTK*     INTEGER   SWITCH TO SELECT TURBULENCE K.E. INITIAL ARRAY (0,1)
C     *INPTS*     INTEGER   SWITCH TO SELECT SUSP. SEDIMENT INITIAL ARRAY (0,1)
C     *INPTZ*     INTEGER   SWITCH TO SELECT SURFACE ELEVATION INITIAL ARRAY (0,1)
C     *INPT2*     INTEGER   SWITCH TO SELECT 2D CURRENT INITIAL ARRAYS (0,1)
C     *INPT3*     INTEGER   SWITCH TO SELECT 3D CURRENT INITIAL ARRAYS (0,1)
C     *INTGS*     INTEGER   SWITCH TO SELECT EXPLICIT/IMPLICIT INTEGRATION (0,1)
C     *IOPTB*     INTEGER   SWITCH TO SELECT BIOLOGICAL MODEL OPTION (0,1)
C     *IOPTC*     INTEGER   SWITCH TO SELECT CONTAMINANT DISPERSION OPTION (0,1)
C     *IOPTD*     INTEGER   SWITCH TO SELECT SALT/HEAT DISPERSION OPTION (0,1,2)
C     *IOPTK*     INTEGER   SWITCH TO SELECT TURBULENCE CLOSURE LEVEL (0,1,2,3)
C     *IOPTM*     INTEGER   SWITCH TO SELECT MET FORCING OPTION (0,1)
C     *IOPTS*     INTEGER   SWITCH TO SELECT SEDIMENT TRANSPORT OPTION (0,1)
C     *IOPTT*     INTEGER   SWITCH TO SELECT TIDAL FORCING (0,1)
C     *IOPTW*     INTEGER   SWITCH TO SELECT WAVE EFFECTS (0,1)
C     *IOPT3*     INTEGER   SWITCH TO SELECT 2-D/3-D CURRENTS (0,1)
C     *IOUTB*     INTEGER   SWITCH TO SELECT BIOLOGICAL MODEL OUTPUT
C     *IOUTC*     INTEGER   SWITCH TO SELECT CONTAMINANT OUTPUT
C     *IOUTD*     INTEGER   SWITCH TO SELECT SALT/HEAT OUTPUT
C     *IOUTK*     INTEGER   SWITCH TO SELECT TURBULENCE OUTPUT
C     *IOUTS*     INTEGER   SWITCH TO SELECT SEDIMENT OUTPUT
C     *IOUTT*     INTEGER   SWITCH TO SELECT TIDAL OUTPUT
C     *IOUT3*     INTEGER   SWITCH TO SELECT 2-D/3-D CURRENT OUTPUT
C     *IOUTF*     INTEGER   SWITCH TO SELECT ARRAY/TIME-SERIES OUTPUT (0,1,2)
C     *TITLE* CHARACTER*80  TITLE OF MODEL RUN
C
C-----------------------------------------------------------------------
C      include 'sedmnt.inc' *** as used in SUBROUTINES ANALYS, BIOLGY, DENSTY,
C                           OUTPA, OUTPT, SEDT ***
C
C*     COMMON *SEDMNT* - SEDIMENT DATA
C
       COMMON/SEDMNT/ SEDC1(NC,NR,NZ), SEDC2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C
C*     NAME       TYPE       PURPOSE
C      ----       ----       -------
C     *SEDC1*     REAL       SUSPENDED SEDIMENT CONCENTRATION, FRACTION 1
C     *SEDC2*     REAL       SUSPENDED SEDIMENT CONCENTRATION, FRACTION 2
C
C-----------------------------------------------------------------------
C      include 'stress.inc' *** as used in SUBROUTINES BSTRESS, CONTNY, CRRNT2,
C                           CRRNT3, INITC, SEDT, TLENG, TURBEN, VEDDY, WAVIN ***
C
C-----------------------------------------------------------------------
C
C*     COMMON *STRESS* - SURFACE AND BOTTOM STRESSES
C
       COMMON/STRESS/ FS(NC,NR), GS(NC,NR), FB(NC,NR), GB(NC,NR),
      1               CDS(NC,NR), CDB(NC,NR)
C
C-----------------------------------------------------------------------
C
C*     NAME       TYPE       PURPOSE
C      ----       ----       -------
C     *FS*        REAL       EAST COMPONENT OF SURFACE STRESS
C     *GS*        REAL       NORTH COMPONENT OF SURFACE STRESS
C     *FB*        REAL       EAST COMPONENT OF BOTTOM STRESS
C     *GB*        REAL       NORTH COMPONENT OF BOTTOM STRESS
C     *CDS*       REAL       2-D ARRAY OF SURFACE FRICTION COEFFICIENT
C     *CDB*       REAL       2-D ARRAY OF BOTTOM FRICTION COEFFICIENT
C
C-----------------------------------------------------------------------
C      include 'tide.inc' *** as used in SUBROUTINES ANALYS, TIDIN ***
C
C*     COMMON *TIDE* - TIDAL PARAMETERS
C
       COMMON/TIDE/ SIGMA(NCON)
C
C-----------------------------------------------------------------------
C
C*     NAME       TYPE       PURPOSE
C      ----       ----       -------
C     *SIGMA*     REAL       FREQUENCY OF (NCON) TIDAL CONSTITUENTS
C
C-----------------------------------------------------------------------
C      include 'time.inc' *** as used in SUBROUTINES ADVDIS, ANALYS, BIOLGY,
C                         BOUNDC, BSTRESS, CONTNY, CRRNT2, CRRNT3, DISSIP, HEAT,
C                         INITC, INCDAT, INTEGR, MASS, METIN, OUTPA, OUTPT,
C                         PRINT, SALT, SEDT, TIDIN, TLENG, TURBEN, VEDDY, WAVIN *
C
C*     COMMON *TIME* - TIME VARIABLES
C
       COMMON/TIME/ DT, DELT, DELM, DELW, DELC, DELS, DELP, DELO, DEL3,
      1             DELD, HOUR, IDATE, IYEAR, IBDATE, IEDATE, NT, NSTEP
C
C-----------------------------------------------------------------------
```

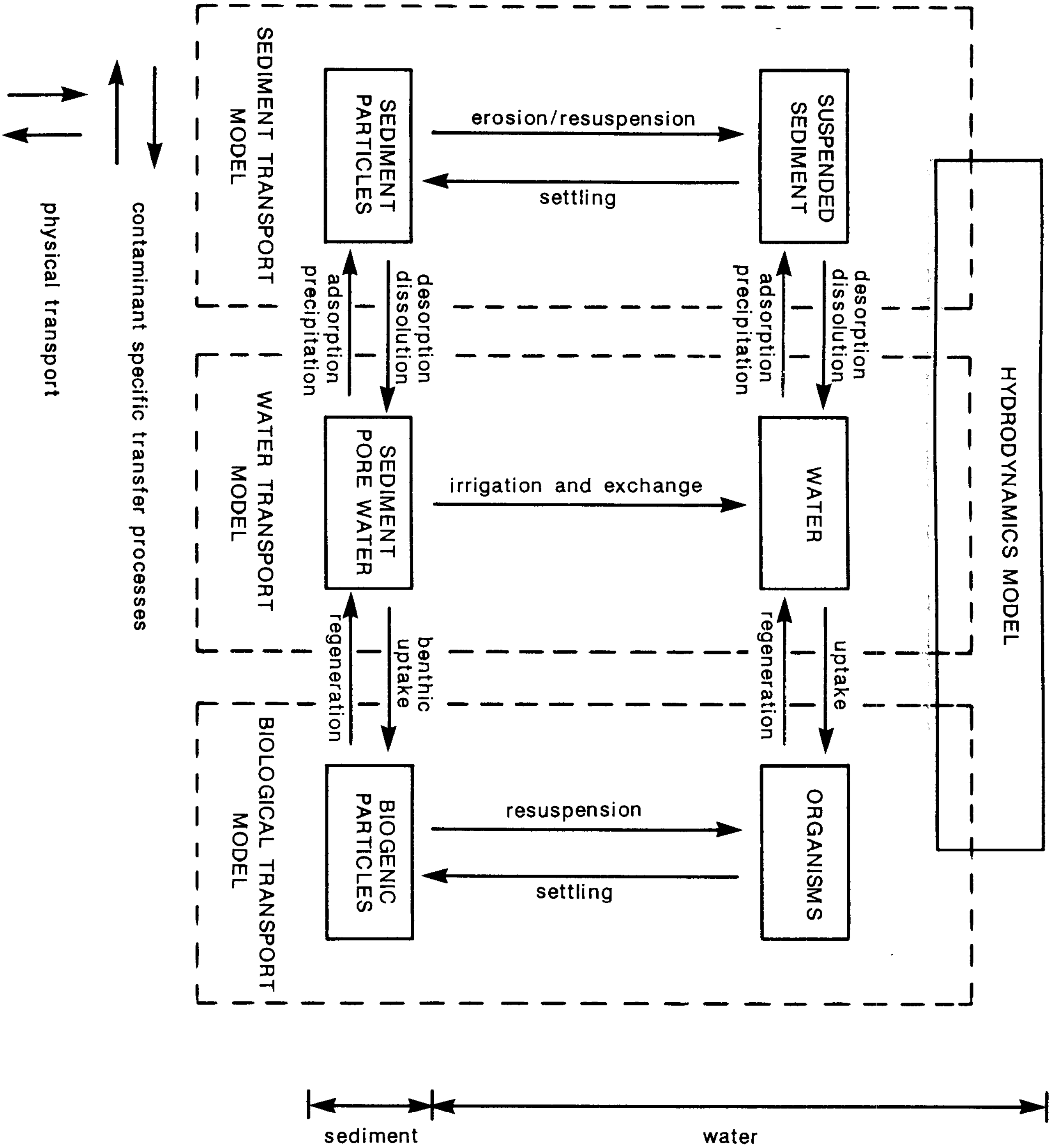


Figure 1 *Conceptual model for a water quality model, from GESAMP Working Group 25.*

```
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C     1. INITIALISATION OF MODEL PARAMETERS AND CONSTANTS
C     --------------------------------------------------
C
C      1.1 READ RUN CONTROL PARAMETERS AND OPTIONS FROM UNIT 5
C      ---------------------------------------------------
C
      READ(5,'(A80)') TITLE
      READ(5,*) IOPTB, IOPTS, IOPTK, IOPT3, IOPTC, IOPTD, IOPTW, IOPTM,
     1          IOPTT
      READ(5,*) INPTB, INPTS, INPTK, INPTC, INPTD, INPT3, INPT2, INPTZ
      READ(5,*) IOUTB, IOUTS, IOUTK, IOUT3, IOUTC, IOUTD, IOUTT, IOUTF
      READ(5,*) IGTRH, IGTRV, INTGS, IADVC, IBOUN
      READ(5,*) DELT, DEL3, DELD, DELM, DELW, DELC, DELS, DELP, DELO
      READ(5,*) IBDATE, IEDATE
C
C      1.2 SET UP COUNTERS FOR INCLUSION OF VARIOUS PROCESSES
C      --------------------------------------------------
C
      IC3D  = DEL3/DELT
      ICMET = DELM/DELT
      ICDEN = DELD/DELT
      ICDIS = DELC/DELT
      ICWAV = DELW/DELT
      ICSED = DELS/DELT
      ICBIO = DELP/DELT
      ICOUT = DELO/DELT
C
C      1.3 CALL INITIALISATION SUBROUTINE
C      ---------------------------------
C
      CALL INITC
C
C      1.4 INPUT TIDAL BOUNDARY AND INITIAL MET DATA IF REQUIRED
C      -----------------------------------------------------
C
      IF (IOPTT.EQ.1) CALL TIDIN
C
      IF (IOPTM.EQ.1) CALL METIN
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C     2. START TIME-STEPPING
C     ---------------------
C
      DO 99 NT = 1, NSTEP
C
C      2.1 TREAT BOUNDARY CONDITIONS
C      ----------------------------
C
      CALL BOUNDC
C
C      2.2 SOLVE EQUATIONS
C      ------------------
C
C       2.21 SOLVE CONTINUITY EQUATION
C       -----------------------------
C
       CALL CONTNY
C
C       2.22 SOLVE 2-D AND 3-D MOMENTUM EQUATIONS
C       ----------------------------------------
C
       CALL CRRNT2
       IF (IOPT3.EQ.1) CALL CRRNT3
C
C       2.23 SOLVE CONSERVATION OF SALT EQUATION EVERY ICDEN TIME-STEPS
C       -------------------------------------------------------------
C
       IF (IOPTD.EQ.1) THEN
        IF (NT/ICDEN*ICDEN.EQ.NT) CALL SALT
       END IF
C
C       2.24 SOLVE CONSERVATION OF HEAT EQUATION EVERY ICDEN TIME-STEPS
C       -------------------------------------------------------------
C
       IF (IOPTD.EQ.1) THEN
        IF (NT/ICDEN*ICDEN.EQ.NT) CALL HEAT
       END IF
C
C       2.25 RECOMPUTE DENSITY FIELDS
C       ----------------------------
```

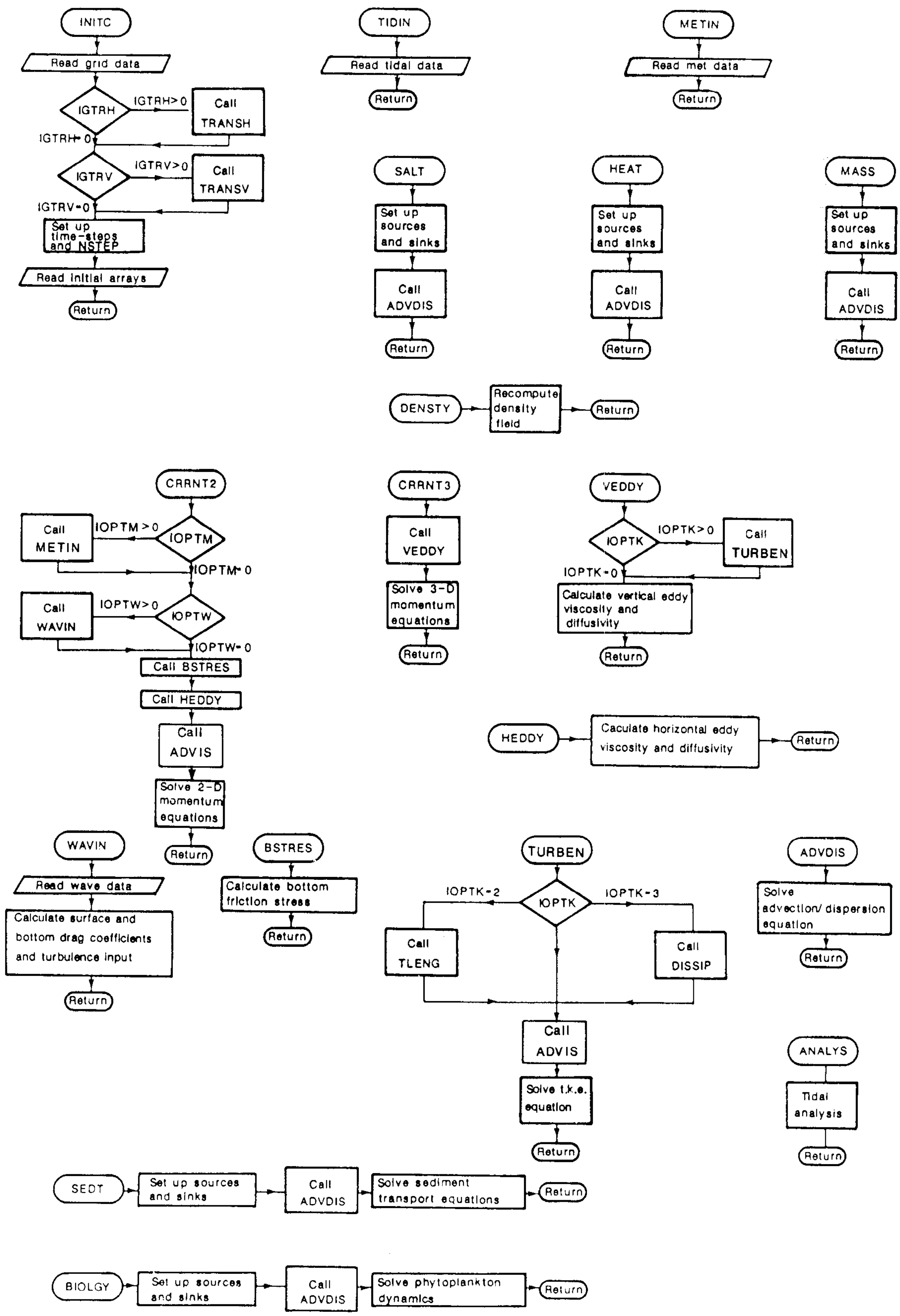


Figure 3 *Detailed subroutine structure.*

```
      include 'visc.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME       TYPE       PURPOSE
C     ----       ----       -------
C    *VAR1*      REAL       LOCAL VARIABLE
C    *VAR2*      REAL       LOCAL VARIABLE
C    *Z1*        REAL       LOCAL ARRAY
C    *Z2*        REAL       LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE ANALYS
C***********************************************************************
C
C    *ANALYS*      TIDAL ANALYSIS
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - PERFORMS TIDAL ANALYSIS ON MODEL OUTPUT ARRAYS
C                     FOR ELEVATIONS AND CURRENTS
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'elev.inc'
      include 'runp.inc'
      include 'sedmnt.inc'
      include 'tide.inc'
      include 'time.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME       TYPE       PURPOSE
C     ----       ----       -------
C    *VAR1*      REAL       LOCAL VARIABLE
C    *VAR2*      REAL       LOCAL VARIABLE
C    *Z1*        REAL       LOCAL ARRAY
C    *Z2*        REAL       LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE BIOLGY
C***********************************************************************
C
C    *BIOLGY*      SOLVE PHYTOPLANKTON DYNAMICS
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - MODELS GROWTH, DECAY AND MOVEMENT OF BIOLOGICAL
C                     VARIABLES: NUTRIENTS, PHYTOPLANKTON, DETRITUS ETC.
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - ADVDIS
C
C***********************************************************************
      include 'param.inc'
```

APPENDIX - Sample FORTRAN Code

```
C***********************************************************************
C
C**    FRAMEWORK FOR NEARSHORE FINE-RESOLUTION MODEL
C      INITIAL APPLICATIONS:
C         1) NORTH SEA PROJECT
C         2) MAST 25 PROJECT, MODELLING OF THE RHINE PLUME
C
C      PHYSICS OPTIONS INCLUDE:
C         TIDES, WAVES, WIND-DRIVEN 3D CURRENTS, TURBULENCE
C         ADVECTION-DISPERSION OF TEMPERATURE AND SALINITY
C         ADVECTION-DISPERSION OF SOLUTES
C         SEDIMENT TRANSPORT
C         PHYTOPLANKTON DYNAMICS
C
C      THE AIM OF THIS OUTLINE PROGRAM IS TO PROVIDE A CONSISTENT
C      AND FLEXIBLE FRAMEWORK WITHIN WHICH TO DEVELOP COMPATIBLE
C      MODULES IN A SUBROUTINE FORMAT. IT IS EXPECTED THAT PARTICULAR
C      APPLICATIONS MAY NOT REQUIRE ALL THE OPTIONS AND THEREFORE
C      WILL USE ONLY PART OF THE CODE, FOR EFFICIENCY. ANY MACHINE-
C      DEPENDENT CODE IS AVOIDED, AND STANDARD FORTRAN 77 IS USED.
C      STREAMLINING FOR MAXIMUM SPEED IS THE RESPONSIBLITY OF THE
C      END-USER.
C
C                  J.WOLF, P.O.L., JULY 1990
C                          modified MAY 1991
C
C***********************************************************************
C
C      THE TYPE OF MODEL HAS BEEN CHOSEN AS A FINITE DIFFERENCE,
C      FORWARD TIME-STEPPING, PRIMITIVE EQUATION MODEL. IT IS ANTICIPATED
C      THAT SOME COORDINATE TRANSFORMATIONS WILL BE REQUIRED SO THAT
C      THIS SHOULD BE INCLUDED AS AN OPTION FOR BOTH THE HORIZONTAL
C      (SPATIAL) AND VERTICAL DIMENSIONS. HOWEVER, TIME-VARYING
C      HORIZONTAL GRID-SCHEMES ARE NOT CATERED FOR. DYNAMIC NESTING OF
C      FINER-RESOLUTION SUB-GRIDS MAY BE REQUIRED. VARIOUS GRID-SCHEMES
C      MAY BE PREFERRED FOR DIFFERENT APPLICATIONS, IN PARTICULAR THE
C      ARAKAWA B AND C GRIDS, SO THIS WILL BE ALLOWED FOR.
C
C***********************************************************************
C
C      DATA TRANSFER PROTOCOL
C      ----------------------
C
C  1)  THE ARRAY-DIMENSIONS ARE TO BE SPECIFIED IN A PARAMETER STATEMENT.
C  2)  OTHER CONSTANTS AND OPTIONS FOR DIFFERENT CASE STUDIES
C      ARE READ IN AT RUN TIME (UNIT 5).
C  3)  ALL TIME-STEPS ARE EXPRESSED AS A MULTIPLE OF THE EXPLICIT COURANT-
C      FRIEDRICHS-LEWY MAXIMUM TIME-STEP WHICH IS READ IN (UNIT 5) AND
C      ALSO CALCULATED FROM THE SPECIFICATION OF THE GRID, AS A CHECK.
C      HOWEVER, INTEGRATION SCHEMES MAY BE IMPLICIT USING MULTIPLES OF THIS,
C      WHICH DEFINES THE COURANT NUMBER OF EACH SCHEME.
C  4)  THE MODEL GRID WILL IN GENERAL BE SET UP IN A PREPROCESSING
C      PROGRAM. GRID-DATA AND BATHYMETRY ARE READ IN ON UNIT 1.
C  5)  INITIAL CONDITIONS ARE READ IN ON UNIT 2.
C  6)  BOUNDARY CONDITIONS ARE READ IN ON UNIT 3.
C  7)  MET. FORCING DATA ARE READ IN ON UNIT 4.
C  8)  MODEL REPORTS ARE WRITTEN ON UNIT 6 (PRINTOUT).
C  9)  FINAL ARRAYS ARE WRITTEN ON UNIT 7 (FOR RESTART).
C 10)  OTHER OUTPUT ARE WRITTEN ON UNITS 10-20.
C 11)  VARIABLES ARE PASSED BETWEEN SUBROUTINES IN GENERAL IN LABELLED
C      COMMON BLOCKS.
C      N.B. THIS CHOICE MAY NOT BE THE FINAL ONE, THE OTHER OPTION BEING TO
C      DYNAMICALLY INITIALISE ALL ARRAYS, SUPPLYING ONE PARAMETER STATEMENT
C      FOR THE MAIN PROGRAM, AND PASSING ARRAY NAMES AND DIMENSIONS AS
C      SUBROUTINE ARGUMENTS. LABELLED COMMON WOULD THEN BE REPLACED BY
C      DIMENSION STATEMENTS
C      THIS MAY BE MORE CLUMSY, BUT HAS ADVANTAGES FOR VECTORISATION.
C 12)  ALL INPUT AND OUTPUT OF MODEL ARRAYS AND TIME-SERIES ARE TO BE
C      FORMATTED.
C 13)  M.K.S. UNITS ARE TO BE USED THROUGHOUT.
C 14)  MODEL ARRAYS ARE TO BE READ/WRITTEN FROM THE BOTTOM LEFT (SOUTHWEST)
C      CORNER, ROW BY ROW. SEPARATE SUBSCRIPTS ARE USED FOR EACH
C      DIMENSION. TO USE THE NORMAL CONVENTION FOR CONTIGUOUS DATA
C      STORAGE A (HORIZONTAL) SPATIAL ARRAY WILL BE WRITTEN E.G.
C      A(NC,NR), WHERE NC = NO. OF COLUMNS (W-E) AND NR = NO. OF ROWS (N-S)
C 15)  THE VERTICAL DIMENSION MAY BE REGARDED AS AN OPTION AND THUS
C      THE OPTIMUM ARRANGEMENT IS FOR THIS TO BE THE 3RD DIMENSION.
C      COUNTING IS FROM THE BOTTOM (SEA-BED) UPWARDS.
C      VECTORISATION, IF IMPLEMENTED, WILL THEREFORE BE OVER THE HORIZONTAL
C      DIMENSIONS. A 3-DIMENSIONAL ARRAY WOULD BE WRITTEN B(NC,NR,NZ),
C      WHERE NZ IS THE NO. OF GRID-BOXES IN THE VERTICAL.
C
C     N.B. MUCH OF THE PROGRAMMING STANDARDS FOLLOW THE 'DOCTOR' SYSTEM
C          AS USED BY THE E.C.M.W.F., REF: GIBSON(1982).
C          HOWEVER THIS IS NOT USED FOR SUBROUTINES AND VARIABLE NAMES,
C          WHICH ARE KEPT AS CLOSE TO COMMONLY USED NOTATION AS POSSIBLE.
C
C***********************************************************************
```

```
C     *BSTRES*      BOTTOM STRESS CALCULATIONS
C
C        AUTHOR -
C
C        LAST UPDATE -
C
C        DESCRIPTION - CALCULATES BOTTOM STRESS FOR 2-D AND 3-D (SLIP)
C                      MOMENTUM EQUATIONS
C
C        REFERENCE -
C
C        CALLING PROGRAM - CRRNT2
C
C        EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'stress.inc'
      include 'time.inc'
      include 'visc.inc'
      include 'waves.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE CONTNY
C***********************************************************************
C
C     *CONTNY*      SOLVE CONTINUITY EQUATION FOR SURFACE ELEVATIONS
C
C        AUTHOR -
C
C        LAST UPDATE -
C
C        DESCRIPTION - SOLVES CONTINUITY BY OPTIONAL INTEGRATION SCHEME
C                      E.G. EXPLICIT, S.O.R, A.D.I (SELECTED BY INTGS)
C
C        REFERENCE -
C
C        CALLING PROGRAM - MAIN
C
C        EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'stress.inc'
      include 'time.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
```

```
C     *NZ*        INTEGER   NO. OF GRID-BOXES IN THE VERTICAL
C     *NL*        INTEGER   NO. OF LEVELS FOR BIOLOGICAL MODEL
C     *NOBZ*      INTEGER   NO. OF OPEN BOUNDARY Z-POINTS
C     *NOBU*      INTEGER   NO. OF OPEN BOUNDARY U-POINTS
C     *NOBV*      INTEGER   NO. OF OPEN BOUNDARY V-POINTS
C     *NCON*      INTEGER   NO. OF TIDAL CONSTITUENTS
C
C-----------------------------------------------------------------------
C      include 'bounds.inc' *** used in SUBROUTINES BOUNDC, TIDIN ***
C
C*     COMMON *BOUNDS* - OPEN BOUNDARY DATA
C
       COMMON/BOUNDS/ HCOSG(NOBZ,NCON), HSING(NOBZ,NCON),
     1                UCOSG(NOBU,NCON), USING(NOBU,NCON),
     2                VCOSG(NOBV,NCON), VSING(NOBV,NCON),
     3                FLOWU(NOBU), FLOWV(NOBV),
     4                SINB(NOBZ), TINB(NOBZ),
     5                CIN1(NOBZ), CIN2(NOBZ),
     6                SEDIN1(NOBZ), SEDIN2(NOBZ),
     7                PPIN1(NOBZ), PPIN2(NOBZ)
C
C-----------------------------------------------------------------------
C
C*     NAME       TYPE       PURPOSE
C      ----       ----       -------
C     *HCOSG*     REAL       AMPLITUDE*COS(PHASE) FOR TIDAL ELEVATION
C     *HSING*     REAL       AMPLITUDE*SIN(PHASE) FOR TIDAL ELEVATION
C     *UCOSG*     REAL       AMPLITUDE*COS(PHASE) FOR TIDAL U-CURRENT
C     *USING*     REAL       AMPLITUDE*SIN(PHASE) FOR TIDAL U-CURRENT
C     *VCOSG*     REAL       AMPLITUDE*COS(PHASE) FOR TIDAL V-CURRENT
C     *VSING*     REAL       AMPLITUDE*SIN(PHASE) FOR TIDAL V-CURRENT
C     *FLOWU*     REAL       RESIDUAL FLOW AT OPEN BOUNDARY U-POINT
C     *FLOWV*     REAL       RESIDUAL FLOW AT OPEN BOUNDARY V-POINT
C     *SIN*       REAL       CONCENTRATION OF CONTAMINANT 1 AT O.B.Z-POINT
C     *TIN*       REAL       CONCENTRATION OF CONTAMINANT 2 AT O.B.Z-POINT
C     *CIN1*      REAL       CONCENTRATION OF CONTAMINANT 1 AT O.B.Z-POINT
C     *CIN2*      REAL       CONCENTRATION OF CONTAMINANT 2 AT O.B.Z-POINT
C     *SEDIN1*    REAL       CONCENTRATION OF SUS. SED. FRACTION 1 AT O.B.Z-POINT
C     *SEDIN2*    REAL       CONCENTRATION OF SUS. SED. FRACTION 2 AT O.B.Z-POINT
C     *PPIN1*     REAL       CONCENTRATION OF PHYTOPLANKTON TYPE 1 AT O.B.Z-POINT
C     *PPIN2*     REAL       CONCENTRATION OF PHYTOPLANKTON TYPE 2 AT O.B.Z-POINT
C
C-----------------------------------------------------------------------
C      include 'concn.inc' *** as used in SUBROUTINES ADVDIS, BIOLGY, DENSTY,
C                          HEAT, MASS, OUTPA, OUTPT, SALT, SEDT ***
C
C*     COMMON *CONCN* - CONCENTRATIONS OF VARIOUS SUBSTANCES
C
       COMMON/CONCN/ S(NC,NR,NZ), T(NC,NR,NZ), RO(NC,NR,NZ),
     1               CONC1(NC,NR,NZ), CONC2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C
C*     NAME       TYPE       PURPOSE
C      ----       ----       -------
C     *S*         REAL       SALINITY
C     *T*         REAL       TEMPERATURE
C     *RO*        REAL       DENSITY
C     *CONC1*     REAL       CONCENTRATION OF CONTAMINANT 1
C     *CONC2*     REAL       CONCENTRATION OF CONTAMINANT 2
C
C-----------------------------------------------------------------------
C      include 'consts.inc' *** as used in all SUBROUTINES ***
C
C*     COMMON *CONSTS* - UNIVERSAL CONSTANTS USED IN PROGRAM
C
       COMMON/CONSTS/ PI, G, CORIOL(NC,NR)
C
C-----------------------------------------------------------------------
C
C*     NAME       TYPE       PURPOSE
C      ----       ----       -------
C     *PI*        REAL       PI=3.141592 (=ASIN(1.)*2.)
C     *G*         REAL       GRAVITATIONAL CONSTANT (9.81 M**2/S)
C     *CORIOL*    REAL       CORIOLIS PARAMETER (DEPENDENT ON LATITUDE)
C
C-----------------------------------------------------------------------
C      include 'count.inc' *** as used in SUBROUTINES BIOLGY, CRRNT2, HEAT,
C                          INTEGR, MASS, METIN, SALT, TURBEN ***
C
C*     COMMON *COUNT* - COUNTERS FOR MULTIPLE TIME-STEPS
C
       COMMON/COUNT/ IC3D, ICBIO, ICDEN, ICDIS, ICMET, ICOUT, ICSED,
     1               ICWAV
C
C-----------------------------------------------------------------------
C
C*     NAME       TYPE       PURPOSE
```

```
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'met.inc'
      include 'runp.inc'
      include 'stress.inc'
      include 'time.inc'
      include 'visc.inc'
      include 'waves.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE DENSTY
C***********************************************************************
C
C    *DENSTY*      CALCULATE DENSITY
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - RECOMPUTES DESITY FIELD BASED ON SALINITY, TEMPERATURE,
C                     AND POSSIBLY CONCENTRATIONS OF CONTAMINANT AND
C                     SUSPENDED SEDIMENT.
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'concn.inc'
      include 'consts.inc'
      include 'runp.inc'
      include 'sedmnt.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE DISSIP
C***********************************************************************
C
C    *DISSIP*      CALCULATE TURBULENCE DISSIPATION RATE
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - SOLVES 2ND EQUATION IN 2-EQUATION TURBULENCE CLOSURE
C                     FOR TURBULENCE DISSIPATION RATE
C                     (OTHER OPTION IS LENGTH-SCALE EQUATION).
C
```

```
C     *GY1*       REAL        STORAGE OF TRANSFORM DATA E.G. DERIVATIVES
C     *GZ1*       REAL        STORAGE OF TRANSFORM DATA E.G. DERIVATIVES
C     *GX2*       REAL        STORAGE OF TRANSFORM DATA E.G. DERIVATIVES
C     *GY2*       REAL        STORAGE OF TRANSFORM DATA E.G. DERIVATIVES
C     *GZ2*       REAL        STORAGE OF TRANSFORM DATA E.G. DERIVATIVES
C     *IN1*       INTEGER     STORAGE OF TRANSFORM DATA E.G. NEAREST NEIGHBOUR
C     *IN2*       INTEGER     STORAGE OF TRANSFORM DATA E.G. NEAREST NEIGHBOUR
C     *IN3*       INTEGER     STORAGE OF TRANSFORM DATA E.G. NEAREST NEIGHBOUR
C     *IN4*       INTEGER     STORAGE OF TRANSFORM DATA E.G. NEAREST NEIGHBOUR
C     *DELX*      REAL        E-W GRID SIZE (NORMALISED, TRANSFORMED COORDINATES)
C     *DELY*      REAL        N-S GRID SIZE (NORMALISED, TRANSFORMED COORDINATES)
C     *DELZ*      REAL        GRID SIZE (NORMALISED, TRANSFORMED COORDINATES)
C                             VERTICAL DIRECTION (POSITIVE UPWARDS)
C     *IGRID*     INTEGER     GRID TYPE (E.G. ARAKAWA A, B, C)
C
C-----------------------------------------------------------------------
C      include 'met.inc' *** as used in SUBROUTINES BIOLGY, CRRNT2, CRRNT3,
C                        HEAT, METIN, OUTPA, OUTPT, TLENG, TURBEN, VEDDY, WAVIN *
C
C*     COMMON *MET* - METEOROLOGICAL FORCING DATA
C
       COMMON/MET/ P1(NC,NR), P2(NC,NR), WINDU1(NC,NR), WINDU2(NC,NR),
     1             WINDV1(NC,NR), WINDV2(NC,NR), SST1(NC,NR),
     2             SST2(NC,NR), SAT1(NC,NR), SAT2(NC,NR), RAD1(NC,NR),
     3             RAD2(NC,NR), HUM1(NC,NR), HUM2(NC,NR)
C
C-----------------------------------------------------------------------
C
C*     NAME       TYPE        PURPOSE
C      ----       ----        -------
C     *HUM1*      REAL        HUMIDITY AT LOWER TIME LEVEL
C     *HUM2*      REAL        HUMIDITY AT HIGHER TIME LEVEL
C     *P1*        REAL        ATMOSPHERIC PRESSURE AT LOWER TIME-LEVEL
C     *P2*        REAL        ATMOSPHERIC PRESSURE AT HIGHER TIME-LEVEL
C     *RAD1*      REAL        DOWNWELLING IRRADIANCE AT LOWER TIME LEVEL
C     *RAD2*      REAL        DOWNWELLING IRRADIANCE AT HIGHER TIME LEVEL
C     *SAT1*      REAL        SURFACE AIR TEMPERATURE AT LOWER TIME LEVEL
C     *SAT2*      REAL        SURFACE AIR TEMPERATURE AT HIGHER TIME LEVEL
C     *SST1*      REAL        SEA SURFACE TEMPERATURE AT LOWER TIME LEVEL
C     *SST2*      REAL        SEA SURFACE TEMPERATURE AT HIGHER TIME LEVEL
C     *WINDU1*    REAL        E-COMPONENT OF 10M WIND VELOCITY AT LOWER TIME-LEVEL
C     *WINDU2*    REAL        E-COMPONENT OF 10M WIND VELOCITY AT HIGHER TIME-LEVEL
C     *WINDV1*    REAL        N-COMPONENT OF 10M WIND VELOCITY AT LOWER TIME-LEVEL
C     *WINDV2*    REAL        N-COMPONENT OF 10M WIND VELOCITY AT HIGHER TIME-LEVEL
C
C-----------------------------------------------------------------------
C      include 'plnktn.inc' *** as used in SUBROUTINES BIOLGY, OUTPA, OUTPT ***
C
C*     COMMON *PLNKTN* - BIOLOGICAL DATA
C
       COMMON/PLNKTN/ PP1(NC,NR,NL), PP2(NC,NR,NL), DO(NC,NR,NL),
     1                RN1(NC,NR,NL), RN2(NC,NR,NL)
C
C-----------------------------------------------------------------------
C
C*     NAME       TYPE        PURPOSE
C      ----       ----        -------
C     *DO*        REAL        DISSOLVED OXYGEN CONCENTRATION
C     *PP1*       REAL        PHYTOPLANKTON PARTICLE CONCENTRATION, TYPE 1
C     *PP2*       REAL        PHYTOPLANKTON PARTICLE CONCENTRATION, TYPE 2
C     *RN1*       REAL        NUTRIENT CONCENTRATION, TYPE 1
C     *RN2*       REAL        NUTRIENT CONCENTRATION, TYPE 2
C
C-----------------------------------------------------------------------
C      include 'runp.inc' *** as used in SUBROUTINES ADVDIS, ANALYS, BIOLGY,
C                         BOUNDC, BSTRESS, CONTNY, CRRNT2, CRRNT3, DENSTY,
C                         DISSIP, HEAT, INITC, INTEGR, MASS, METIN, PRINT, SALT,
C                         SEDT, TLENG, TURBEN, VEDDY ***
C
C*     COMMON *RUNP* - MODEL RUN PARAMETERS
C
       COMMON/RUNP/ IOPTB, IOPTS, IOPTK, IOPT3, IOPTC, IOPTW, IOPTD,
     1              IOPTM, IOPTT, INPTB, INPTS, INPTK, INPTC, INPTD,
     2              INPT3, INPT2, INPTZ, IOUTB, IOUTS, IOUTK, IOUT3,
     3              IOUTC, IOUTD, IOUTT, IOUTF, IGTRH, IGTRV, INTGS,
     4              IADVC, IBOUN, TITLE
       CHARACTER*80 TITLE
C
C-----------------------------------------------------------------------
C
C*     NAME       TYPE        PURPOSE
C      ----       ----        -------
C     *IADVC*     INTEGER     SWITCH TO SELECT ADVECTION SCHEME (0,1,2,...)
C     *IBOUN*     INTEGER     SWITCH TO SELECT BOUNDARY CONDITIONS (0,1,2,...)
C     *IGTRH*     INTEGER     SWITCH TO SELECT HORIZONTAL COORDINATE TRANSFORM (0,1)
C     *IGTRV*     INTEGER     SWITCH TO SELECT VERTICAL COORDINATE TRANSFORM (0,1)
C     *INPTB*     INTEGER     SWITCH TO SELECT BIOLOGICAL INITIAL ARRAYS (0,1)
C     *INPTC*     INTEGER     SWITCH TO SELECT CONTAMINANT INITIAL ARRAYS (0,1)
```

```
      SUBROUTINE HEDDY
C***********************************************************************
C
C    *HEDDY*       SET UP HORIZONTAL EDDY VISCOSITY, DIFFUSIVITY
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - CALCULATES HORIZONTAL EDDY VISCOSITY AND DIFFUSIVITY
C                     WHICH MAY DEPEND ON GRID SIZE AND FLOW PARAMETERS
C
C       REFERENCE -
C
C       CALLING PROGRAM - CRRNT2
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'visc.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE INITC
C***********************************************************************
C
C    *INITC*       SET UP INITIAL CONDITIONS
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - 1) READ MODEL BATHYMETRY AND GRID DATA FROM UNIT 1
C                     2) PERFORM COORDINATE TRANSFORMATIONS IF REQUIRED
C                     3) CALCULATE CFL LIMIT ON TIME-STEP (CHECK DT)
C                     4) CALCULATE TOTAL NO. OF TIME-STEPS (NSTEP)
C                     5) READ INITIAL ARRAYS IF SUPPLIED, ON UNIT 2,
C                        SELECTED BY INPTB, INPTS, INPTK, INPTC, INPTD,
C                        INPT3, INPT2, INPTZ
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - TRANSH, TRANSV
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'stress.inc'
      include 'time.inc'
      include 'visc.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
```

```
C
C*     NAME        TYPE       PURPOSE
C      ----        ----       -------
C      *DT*        REAL       EXPLICIT CFL TIME-STEP
C                             (MAY DIFFER FROM DT, DELT<=DT FOR EXPLICIT CODE)
C      *DEL3*      REAL       TIME-STEP FOR 3D CURRENTS
C      *DELC*      REAL       TIME-STEP FOR ADVECTION-DIFFUSION
C      *DELD*      REAL       TIME-STEP FOR UPDATE OF DENSITY FIELD
C      *DELM*      REAL       MET. INPUT TIME-STEP (INTEGER MULTIPLE OF DELT)
C      *DELO*      REAL       TIME-STEP FOR PHYTOPLANKTON DYNAMICS *
C      *DELP*      REAL       TIME-STEP FOR PHYTOPLANKTON DYNAMICS *
C      *DELS*      REAL       TIME-STEP FOR SEDIMENT TRANSPORT     *
C      *DELT*      REAL       HYDRODYNAMIC MODEL TIME-STEP
C      *DELW*      REAL       WAVE INPUT TIME-STEP (INTEGER MULTIPLE OF DELT)
C      *HOUR*      REAL       RUN TIME IN HOURS FROM START TIME = 0
C      *IBDATE*    INTEGER    BEGIN DATE (MMDDHHMM - MONTH, DAY, HOUR, MINUTE)
C      *IDATE*     INTEGER    DATE (MMDDHHMM - MONTH, DAY, HOUR, MINUTE)
C      *IEDATE*    INTEGER    END DATE (MMDDHHMM - MONTH, DAY, HOUR, MINUTE)
C      *IYEAR*     INTEGER    YEAR
C      *NT*        INTEGER    TIME-STEP COUNTER
C      *NSTEP*     INTEGER    TOTAL NO. OF TIME-STEPS
C
C-----------------------------------------------------------------------
C      include 'turbke.inc' *** as used in SUBROUTINES ADVDIS, BIOLGY, DISSIP,
C                               OUTPA, OUTPT, SEDT, TLENG, TURBEN, VEDDY ***
C
C*     COMMON *TURBKE* - TURBULENCE DATA
C
       COMMON/TURBKE/ TKE(NC,NR,NZ), ZL(NZ), DISS(NC,NR,NZ), Z0(NC,NR)
C
C-----------------------------------------------------------------------
C
C*     NAME        TYPE       PURPOSE
C      ----        ----       -------
C      *DISS*      REAL       3-D ARRAY OF TURBULENCE DISSIPATION RATE
C      *TKE*       REAL       3-D ARRAY OF TURBULENCE KINETIC ENERGY DENSITY
C      *ZL*        REAL       1-D (VERTICAL) ARRAY OF TURBULENCE LENGTH SCALE
C      *Z0*        REAL       2-D (HORIZONTAL) ARRAY OF ROUGHNESS LENGTHS
C
C-----------------------------------------------------------------------
C      include 'visc.inc' *** as used in SUBROUTINES ADVDIS, BSTRES, CRRNT3,
C                             HEAT, HEDDY, INITC, MASS, SALT, SEDT, TLENG, TURBEN,
C                             VEDDY ***
C
C*     COMMON *VISC* - EDDY VISCOSITIES, EDDY DIFFUSIVITIES
C
       COMMON/VISC/ HEDDYV(NC,NR), VEDDYV(NC,NR,NZ),
      1             HEDDYD(NC,NR), VEDDYD(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C
C*     NAME        TYPE       PURPOSE
C      ----        ----       -------
C      *HEDDYD*    REAL       HORIZONTAL EDDY DIFFUSIVITY
C      *HEDDYV*    REAL       HORIZONTAL EDDY VISCOSITY
C      *VEDDYD*    REAL       VERTICAL EDDY DIFFUSIVITY
C      *VEDDYV*    REAL       VERTICAL EDDY VISCOSITY
C
C-----------------------------------------------------------------------
C      include 'waves.inc' *** as used in SUBROUTINES BSTRES, CRRNT2, CRRNT3,
C                              DISSIP, METIN, TLENG, TURBEN, VEDDY, WAVIN ***
C
C*     COMMON *WAVES* - WAVE INPUT DATA
C
       COMMON/WAVES/ HS(NC,NR), TW(NC,NR), WBR(NC,NR)
C
C-----------------------------------------------------------------------
C
C*     NAME        TYPE       PURPOSE
C      ----        ----       -------
C      *HS*        REAL       SIGNIFICANT WAVE HEIGHT
C      *TW*        REAL       MEAN WAVE PERIOD
C      *WBR*       REAL       WAVE BREAKING SOURCE TERM
C
C-----------------------------------------------------------------------
C      include 'worksp.inc' *** as used in all SUBROUTINES ***
C
C*     COMMON *WORKSP* - WORKING STORAGE
C
       COMMON/WORKSP/ TEMP1(NC,NR,NZ), TEMP2(NC,NR,NZ), TEMP3(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C
C*     NAME        TYPE       PURPOSE
C      ----        ----       -------
C      *TEMP1*     REAL       TEMPORARY ARRAY
C      *TEMP2*     REAL       TEMPORARY ARRAY
C      *TEMP3*     REAL       TEMPORARY ARRAY
```

```
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE INTEGR
C***********************************************************************
C
C    *INTEGR*      TIME-AVERAGING OF VARIABLES
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - PERFORMS TIME-INTEGRATION OF VARIABLES FROM
C                     FUNDAMENTAL TIME-STEP AS REQUIRED BY OTHER PROCESSES.
C
C       REFERENCE -
C
C       CALLING PROGRAM - CONTNY, CRRNT2, CRRNT3
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'count.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'time.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE MASS
C***********************************************************************
C
C    *MASS*        MASS TRANSPORT OF CONTAMINANT
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - SOLVES TRANSPORT EQUATION FOR ABITRARY SUBSTANCE
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - ADVDIS
C
C***********************************************************************
      include 'param.inc'
      include 'concn.inc'
      include 'consts.inc'
      include 'count.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'time.inc'
      include 'visc.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
```

```
C
        IF (NT/ICDEN*ICDEN.EQ.NT) CALL DENSTY
C
C       2.26 SOLVE SEDIMENT TRANSPORT EQUATION EVERY ICSED TIME-STEPS
C       -------------------------------------------------------------
C
        IF (IOPTS.EQ.1) THEN
         IF (NT/ICSED*ICSED.EQ.NT) CALL SEDT
        END IF
C
C       2.27 SOLVE MASS TRANSPORT EQUATION EVERY ICDIS TIME-STEPS
C       ---------------------------------------------------------
C
        IF (IOPTC.EQ.1) THEN
         IF (NT/ICDIS*ICDIS.EQ.NT) CALL MASS
        END IF
C
C       2.28 SOLVE PHYTOPLANKTON DYNAMICS EVERY ICBIO TIME-STEPS
C       --------------------------------------------------------
C
        IF (IOPTB.EQ.1) THEN
         IF (NT/ICBIO*ICBIO.EQ.NT) CALL BIOLGY
        END IF
C
C      2.3 TIME-INTEGRATION AND OUTPUT
C      ------------------------------
C
       CALL INTEGR
       IF (IOUTF.GT.0) THEN
        IF (NT/ICOUT*ICOUT.EQ.NT) CALL OUTPA
       END IF
C
   99 CONTINUE
C
C     END OF MAIN PROGRAM LOOP
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C     3. ANALYSIS AND OUTPUT
C     ----------------------
C
      IF (IOPTT.EQ.1) CALL ANALYS
C
      CALL PRINT
      IF (IOUTF.GT.0) CALL OUTPA
      IF (IOUTF.NE.1) CALL OUTPT
C
C     END OF MAIN PROGRAM
C
C-----------------------------------------------------------------------
C
      STOP
      END
C
      SUBROUTINE ADVDIS
C***********************************************************************
C
C**    *ADVDIS*      SOLVES ADVECTION-DISPERSION EQUATION
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - SOLVES THE ADVECTION-DISPERSION EQUATION FOR
C                     AN ARBITRARY WATER-MASS CHARACTERISTIC, CONCENTRATION
C                     OF CONTAMINANT, OR PARTICLES IN SUSPENSION.
C                     OPTION TO SELECT VARIOUS ADVECTION SCHEMES.
C
C       REFERENCE -
C
C       CALLING PROGRAM - CRRNT2, SALT, HEAT, MASS, SEDT, TURBKE, BIOLGY
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'concn.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'sedmnt.inc'
      include 'time.inc'
      include 'turbke.inc'
```

```
      include 'plnktn.inc'
      include 'sedmnt.inc'
      include 'time.inc'
      include 'turbke.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE OUTPT
C***********************************************************************
C
C    *OUTPT*       OUTPUT TIME SERIES
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION -
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'concn.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'met.inc'
      include 'plnktn.inc'
      include 'sedmnt.inc'
      include 'time.inc'
      include 'turbke.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE PRINT
C***********************************************************************
C
C    *PRINT*       OUTPUT REPORT TO UNIT 6
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION -
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
```

```
      include 'concn.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'count.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'met.inc'
      include 'plnktn.inc'
      include 'runp.inc'
      include 'sedmnt.inc'
      include 'time.inc'
      include 'turbke.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE BOUNDC
C***********************************************************************
C
C    *BOUNDC*      TREAT BOUNDARY CONDITIONS
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - APPLIES BOUNDARY CONDITION TO ALL VARIABLES USING
C                     VARIOUS TYPES OF BOUNDARY CONDITION E.G.
C                     1) ELEVATION SPECIFIED
C                     2) RADIATION CONDITIONS
C                     3) SPONGE LAYERS (ENERGY-ABSORBENT) ETC.
C                     SELECT OPTION USING IBOUN
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'bounds.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'time.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE BSTRES
C***********************************************************************
C
```

```
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - ADVDIS
C
C***********************************************************************
      include 'param.inc'
      include 'concn.inc'
      include 'consts.inc'
      include 'count.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'sedmnt.inc'
      include 'stress.inc'
      include 'time.inc'
      include 'turbke.inc'
      include 'visc.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME       TYPE      PURPOSE
C     ----       ----      -------
C    *VAR1*      REAL      LOCAL VARIABLE
C    *VAR2*      REAL      LOCAL VARIABLE
C    *Z1*        REAL      LOCAL ARRAY
C    *Z2*        REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE TIDIN
C***********************************************************************
C
C    *TIDIN*        SET UP TIDAL BOUNDARY CONDITIONS
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION -
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'bounds.inc'
      include 'consts.inc'
      include 'tide.inc'
      include 'time.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME       TYPE      PURPOSE
C     ----       ----      -------
C    *VAR1*      REAL      LOCAL VARIABLE
C    *VAR2*      REAL      LOCAL VARIABLE
C    *Z1*        REAL      LOCAL ARRAY
C    *Z2*        REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE TLENG
C***********************************************************************
C
```

```
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE CRRNT2
C***********************************************************************
C
C    *CRRNT2*      SOLVE 2-D MOMENTUM EQUATIONS
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - SOLVES BAROTROPIC COMPONENT OF CURRENT BY
C                     OPTIONAL INTEGRATION SCHEME (SELECTED BY INTGS)
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - BSTRES, HEDDY, WAVIN, METIN, ADVDIS
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'count.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'met.inc'
      include 'runp.inc'
      include 'stress.inc'
      include 'time.inc'
      include 'waves.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME       TYPE      PURPOSE
C     ----       ----      -------
C    *VAR1*      REAL      LOCAL VARIABLE
C    *VAR2*      REAL      LOCAL VARIABLE
C    *Z1*        REAL      LOCAL ARRAY
C    *Z2*        REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C       1. INPUT NEW MET DATA EVERY ICMET TIME-STEPS
C       -------------------------------------------
C
       IF (IOPTM.EQ.1) THEN
        IF (NT/ICMET*ICMET.EQ.NT) CALL METIN
       ENDIF
      RETURN
      END
C
      SUBROUTINE CRRNT3
C***********************************************************************
C
C    *CRRNT3*      SOLVE 3-D MOMENTUM EQUATIONS
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - SOLVES DEPTH-VARYING PART OF MOMENTUM EQUATIONS
C                     BY SOLUTION METHOD DEPENDING ON INTGS.
C                     MAY INCLUDE BUOYANCY TERMS.
C                     TRANSFORMED EQUATIONS CONTAINING EXTRA TERMS
C                     SELECTED BY IGTRH, IGTRV.
C                     TURBULENCE CLOSURE LEVEL SELECTED BY IOPTK, IN VEDDY.
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - VEDDY
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
```

```
      RETURN
      END
C
      SUBROUTINE TRANSV
C***********************************************************************
C
C    *TRANSV*      TRANSFORM VERTICAL COORDINATE (OPTIONAL)
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION -
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'depths.inc'
      include 'grid.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE TURBEN
C***********************************************************************
C
C    *TURBEN*      SOLVE TURBULENCE ENERGY EQUATIONS (VARIOUS CLOSURES)
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION -
C
C       REFERENCE -
C
C       CALLING PROGRAM - VEDDY
C
C       EXTERNALS - TLENG, DISSIP, WAVIN, ADVDIS
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'count.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'met.inc'
      include 'runp.inc'
      include 'stress.inc'
      include 'time.inc'
      include 'turbke.inc'
      include 'visc.inc'
      include 'waves.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
```

```
C       REFERENCE -
C
C       CALLING PROGRAM - TURBEN
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'time.inc'
      include 'turbke.inc'
      include 'waves.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE HEAT
C***********************************************************************
C
C    *HEAT*         SOLVE HEAT BUDGET
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - SOLVES HEAT BALANCE EQUATION, INCLUDING SURFACE
C                     HEAT FLUX
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - ADVDIS
C
C***********************************************************************
      include 'param.inc'
      include 'concn.inc'
      include 'consts.inc'
      include 'count.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'met.inc'
      include 'runp.inc'
      include 'time.inc'
      include 'visc.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
```

```
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C----------------------------------------------------------------------
C*    NAME        TYPE        PURPOSE
C     ----        ----        -------
C    *VAR1*       REAL        LOCAL VARIABLE
C    *VAR2*       REAL        LOCAL VARIABLE
C    *Z1*         REAL        LOCAL ARRAY
C    *Z2*         REAL        LOCAL ARRAY
C
C----------------------------------------------------------------------
C
C
      RETURN
      END
```

```
C     ----        ----        -------
C    *VAR1*      REAL        LOCAL VARIABLE
C    *VAR2*      REAL        LOCAL VARIABLE
C    *Z1*        REAL        LOCAL ARRAY
C    *Z2*        REAL        LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C     1. READ MODEL BATHYMETRY AND GRID DATA ON UNIT 1.
C     -------------------------------------------------
C
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C     2. TRANSFORMATION OF GRID IF REQUIRED
C     -------------------------------------
C
      IF (IGTRH.EQ.1) CALL TRANSH
C
      IF (IGTRV.EQ.1) CALL TRANSV
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C     3. CALCULATE CFL LIMIT ON TIME-STEP (CHECK DELT)
C     ------------------------------------------------
C
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C     4. CALCULATE TOTAL NO. OF TIME-STEPS (NSTEP)
C     --------------------------------------------
C
      nstep=1
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C
C     5. READ INITIAL ARRAYS IF SUPPLIED, ON UNIT 2
C     ---------------------------------------------
C
C-----------------------------------------------------------------------
C
      write(6,'('' nstep = '',i5)') nstep
      RETURN
      END
C
      SUBROUTINE INCDAT
C***********************************************************************
C
C    *INCDAT*      INCREMENT DATE
C
C      AUTHOR -
C
C      LAST UPDATE -
C
C      DESCRIPTION - RECOMPUTE DATE AND TIME AS REQUIRED
C
C      REFERENCE -
C
C      CALLING PROGRAM - MAIN
C
C      EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'time.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME        TYPE        PURPOSE
C     ----        ----        -------
C    *VAR1*      REAL        LOCAL VARIABLE
C    *VAR2*      REAL        LOCAL VARIABLE
C    *Z1*        REAL        LOCAL ARRAY
C    *Z2*        REAL        LOCAL ARRAY
C
```

```
C
C-----------------------------------------------------------------------
C*    NAME       TYPE       PURPOSE
C     ----       ----       -------
C    *VAR1*      REAL       LOCAL VARIABLE
C    *VAR2*      REAL       LOCAL VARIABLE
C    *Z1*        REAL       LOCAL ARRAY
C    *Z2*        REAL       LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE METIN
C***********************************************************************
C
C    *METIN*       MET FORCING
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - READS IN ATMOSPHERIC INPUT DATA AT SELECTED INTERVALS
C                     INCLUDING WINDS, ATMOS. PRESSURE, SEA AND AIR TEMPERATURE,
C                     PRECIPITATION AS AVAILABLE.
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN, CRRNT2
C
C       EXTERNALS - WAVIN
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'count.inc'
      include 'depths.inc'
      include 'met.inc'
      include 'runp.inc'
      include 'time.inc'
      include 'waves.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME       TYPE       PURPOSE
C     ----       ----       -------
C    *VAR1*      REAL       LOCAL VARIABLE
C    *VAR2*      REAL       LOCAL VARIABLE
C    *Z1*        REAL       LOCAL ARRAY
C    *Z2*        REAL       LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE OUTPA
C***********************************************************************
C
C    *OUTPA*       OUTPUT ARRAYS
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION - WRITES SELECTED OUTPUT ARRAYS TO REQUIRED OUPUT UNITS.
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'concn.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'met.inc'
```

```
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'depths.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'time.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE SALT
C***********************************************************************
C
C    *SALT*         SOLVE SALINITY CONSERVATION EQUATION
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION -
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - ADVDIS
C
C***********************************************************************
      include 'param.inc'
      include 'concn.inc'
      include 'consts.inc'
      include 'count.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'runp.inc'
      include 'time.inc'
      include 'visc.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE SEDT
C***********************************************************************
C
C    *SEDT*         SOLVE SEDIMENT TRANSPORT EQUATIONS (VARIOUS FRACTIONS)
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION -
```

```
C    *TLENG*      SOLVE TURBULENCE ENERGY LENGTH SCALE EQUATIONS
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION -
C
C       REFERENCE -
C
C       CALLING PROGRAM - TURBEN
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'met.inc'
      include 'runp.inc'
      include 'stress.inc'
      include 'time.inc'
      include 'turbke.inc'
      include 'visc.inc'
      include 'waves.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
      RETURN
      END
C
C
      SUBROUTINE TRANSH
C***********************************************************************
C
C    *TRANSH*      TRANSFORM HORIZONTAL COORDINATES (OPTIONAL)
C
C       AUTHOR -
C
C       LAST UPDATE -
C
C       DESCRIPTION -
C
C       REFERENCE -
C
C       CALLING PROGRAM - MAIN
C
C       EXTERNALS - NONE
C
C***********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'depths.inc'
      include 'grid.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C-----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C-----------------------------------------------------------------------
C
C
```

```
C    *Z1*       REAL       LOCAL ARRAY
C    *Z2*       REAL       LOCAL ARRAY
C
C----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE VEDDY
C**********************************************************************
C
C    *VEDDY*       SET UP VERTICAL EDDY VISCOSITY, DIFFUSIVITY
C
C      AUTHOR -
C
C      LAST UPDATE -
C
C      DESCRIPTION -
C
C      REFERENCE -
C
C      CALLING PROGRAM - CRRNT3
C
C      EXTERNALS - TURBEN
C
C**********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'met.inc'
      include 'runp.inc'
      include 'stress.inc'
      include 'time.inc'
      include 'turbke.inc'
      include 'visc.inc'
      include 'waves.inc'
      include 'worksp.inc'
C
C*    LOCAL VARIABLES
C
      DIMENSION Z1(NC,NR,NZ), Z2(NC,NR,NZ)
C
C----------------------------------------------------------------------
C*    NAME      TYPE      PURPOSE
C     ----      ----      -------
C    *VAR1*     REAL      LOCAL VARIABLE
C    *VAR2*     REAL      LOCAL VARIABLE
C    *Z1*       REAL      LOCAL ARRAY
C    *Z2*       REAL      LOCAL ARRAY
C
C----------------------------------------------------------------------
C
C
      RETURN
      END
C
      SUBROUTINE WAVIN
C**********************************************************************
C
C    *WAVIN*       INCLUDE WAVE EFFECTS
C
C      AUTHOR -
C
C      LAST UPDATE -
C
C      DESCRIPTION -
C
C      REFERENCE -
C
C      CALLING PROGRAM - CRRNT2, CRRNT3
C
C      EXTERNALS - NONE
C
C**********************************************************************
      include 'param.inc'
      include 'consts.inc'
      include 'crrnts.inc'
      include 'depths.inc'
      include 'elev.inc'
      include 'grid.inc'
      include 'met.inc'
      include 'stress.inc'
      include 'time.inc'
      include 'waves.inc'
```