Comprehensive Coastal Circulation Simulation using Finite Elements: Nonlinear Prognostic Time-Stepping Model

QUODDY3 User's Manual

Justin T. C. Ip and Daniel R. Lynch

Thayer School of Engineering
Dartmouth College
Hanover, New Hampshire, U.S.A.

May 12 1995

Report Number NML95-1

Abstract

This document is intended to serve as a user's manual and reference for a state-of-the-art finite-element computer simulation program, QUODDY3, for coastal ocean circulation modeling. The model is three-dimensional and fully nonlinear with a free surface, incorporates advanced turbulence closure, and operates in tidal time. Variable horizontal and vertical resolution are facilitated by the use of unstructured meshes of linear triangles in the horizontal, and structured linear elements in the vertical.

This manual describes the mathematical model, the numerical algorithm, and the general structure of the simulation software. The two earlier versions (QUODDY1 and QUODDY2) leading to the current QUODDY3 code are also described briefly. Four examples with step by step protocols are provided to illustrate the usage of the program.

The software is written in ANSI FORTRAN 77 and was developed by the Numerical Methods Laboratory at Dartmouth College. It is generally compatible with the earlier FUNDY series of models. Those provide harmonic diagnostic solutions on the same meshes which can be used as initial concisions for the QUODDY series.

1 Introduction

QUODDY is a family of Fortran realizations of the 3-D Finite Element shelf circulation model described in Lynch *et al.* [1995]. The distinguishing features of QUODDY are the incorporation of a) fully nonlinear hydrodynamics in the time domain; and b) advanced turbulence closure. This model series forms a companion to the linearized diagnostic harmonic models in the FUNDY series [Lynch *et al.*, 1989; Naimie and Lynch, 1993]. To the extent possible, the essential features and structure of the FUNDY models has been preserved in QUODDY. A working familiarity with FUNDY is a prerequisite to the effective use of QUODDY.

QUODDY3 is a free-surface, tide-resolving model based on the conventional 3-D shallow water equations — i.e. the Reynolds-averaged Navier-Stokes equations for an incompressible, hydrostatic, Boussinesq fluid with a free surface, partially mixed vertically, and driven by rotation, wind, tide, and barotropic and baroclinic pressure gradients. The depth-averaged continuity equation is expressed in "Wave Equation" form. The primitive 3-D momentum equation is expressed in "non-conservative" or "force-balance" form. The horizontal coordinate system is Cartesian, on an f-plane. The Galerkin weighted residual method is used to obtain the weak form, which is discretized on simple linear elements (triangular in the horizontal, linear in the vertical) with all variables expressed in the same linear basis. A general terrain-following vertical coordinate is used, with a flexible FEM approach to vertical resolution. This provides continuous tracking of the free surface and proper resolution of surface and bottom boundary layers. Temperature and salinity are transported in tidal time via conventional transport equations and the density field is closed prognostically via an equation of state. Implementation of these features follows the general algorithmic approach of Lynch and Werner [1991], with improvements and extensions accumulated through the FUNDY modeling experience (e.g. Lynch and Naimie [1993]; Naimie *et al.* [1995]).

Vertical mixing of momentum, heat and mass is represented by a level 2.5 turbulence closure scheme [Mellor and Yamada, 1982]. This approach provides stratification- and shear-dependent mixing coefficients which evolve with the simulation and requires the prognostic evaluation of two additional macroscopic state variables to characterize the turbulence distribution. The improvements of Galperin *et al.* [1988] have been found to be very important in avoiding unphysical situations on Georges Bank and have been adopted as standard. In the horizontal, Smagorinsky-type closure provides shear- and mesh scale-dependent eddy viscosity [Smagorinsky, 1963].

QUODDY2 is a reduced version with only a single mass variable (density). QUODDY1 is a constant density version with simplified turbulence closure and no mass transport component.

2 Software Overview

The QUODDY series, thus far, consists of three individual models — QUODDY1, QUODDY2, and QUODDY3, with increasingly more sophisticated numerical implementation and more realistic physics in each subsequent model. These models are FORTRAN 77 implementation of finite element solutions of the 3-D nonlinear shallow water equations, based on the algorithm described in Lynch and Werner [1991]. Solutions are obtained in the time domain for the fluid velocity and sea surface elevation. Each of these models is forced by tide or other barotropic boundary conditions, wind, surface heating, baroclinic pressure gradient, temperature and salinity of the water column, all specified by the user. The Davies and Furnes turbulence closure (DF80) [Davies and Furnes, 1980] is used in the vertical in the QUODDY1 model, and the Mellor-Yamada level 2.5 turbulent closure model (MY25) [Mellor-Yamada, 1974] with improvements of Galperin *et al.* [1988] (GP88) is employed in the vertical in both the QUODDY2 and QUODDY3 models, with a linearized partial-slip condition enforced at the bottom. The horizontal viscosity, either uniform or velocity and mesh-size dependent [Smagorinsky, 1963], sidewall slip coefficient, vertical viscosity coefficient, and bottom stress coefficient are arbitrary and can be included at the discretion of the user.

The model uses a conventional horizontal grid of linear triangles, which must be provided by the user. A 3-D mesh is automatically constructed within QUODDY from the horizontal grid as follows: the horizontal mesh is projected downward to the bottom in perfectly vertical lines, and each line is discretized into the same number of vertical elements. These are then connected horizontally in the identical topology as the original 2-D mesh, thereby filling the volume with 6-node linear elements of prism shape. Effectively, this creates an (x,y,ε) coordinate system (see Figure 1). QUODDY can employ uniform vertical meshing as well as a variety of non-uniform vertical meshing on each vertical line.

The general structure of the QUODDY3 software program is illustrated in Figure 2. Its two predecessor models — QUODDY1 and QUODDY2 have the almost identical structure and will not be presented in detail. In all three models, there are three main programs and an include file for variables dimensioning purposes.

1. *Core program*
   (a) `quoddy1.f` is the core program for the QUODDY1 model which performs all the finite element assembly, solution operations, and time integration, according to the algorithm given by Lynch and Werner [1991] using the DF80 turbulence closure.
   (b) `quoddy2.1.3.f` is the core program for the QUODDY2 model employing MY25 turbulence closure with GP88 improvements the baroclinic forcing is included and density is to evolve in a prognostic sense.
   (c) `quoddy3.1.f` and `quoddy3.2.f` are the core program for the QUODDY3 model versions 1 and 2 respectively, employing MY25 turbulence closure with GP88 improvements with prognostic temperature and salinity. In QUODDY3 version 2, point source terms are also included.

   In all cases, `quoddy1.f`, `quoddy2.1.3f`, `quoddy3.1.f`, and `quoddy3.2.f` read the same formatted input file `quoddy.inq` and writes a formatted echo file with "`.echo`"suffix. The latter contains a summary of all the input data and files and diagnostic data at run time.

2. *Fixed subroutines*
   Each of the core programs must be linked to a supplement program, **`fixsubs.f`**, containing a number of subroutines including global matrix assembler, various banded matrix solvers, horizontal viscosity calculations, turbulence closure schemes, equation of state, baroclinic forcing, etc., in addition to a number of service subroutines providing standard data input and output to be used by the user subroutines if so desired. These fixed subroutines are universal and are called by all the models in the QUODDY series.

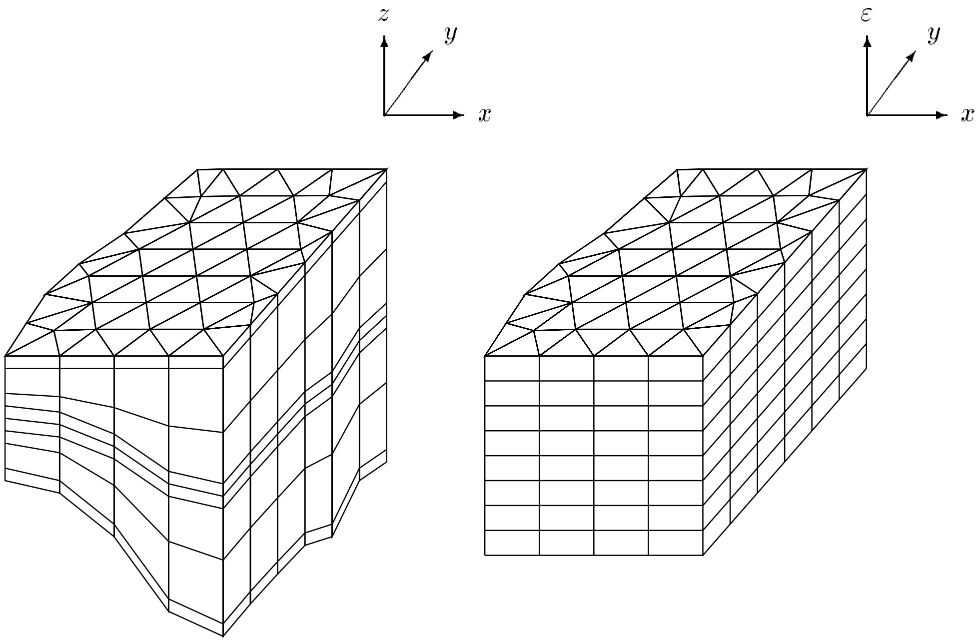

Figure 1: Main features of the 3-D layered mesh (this figure is taken from Naimie and Lynch [1993]): (1) element sides perfectly vertical, (2) variable vertical mesh spacing allows resolution of boundary and internal layers, (3) mesh spacing is uniform in mapped (x,y,ε) coordinate system.

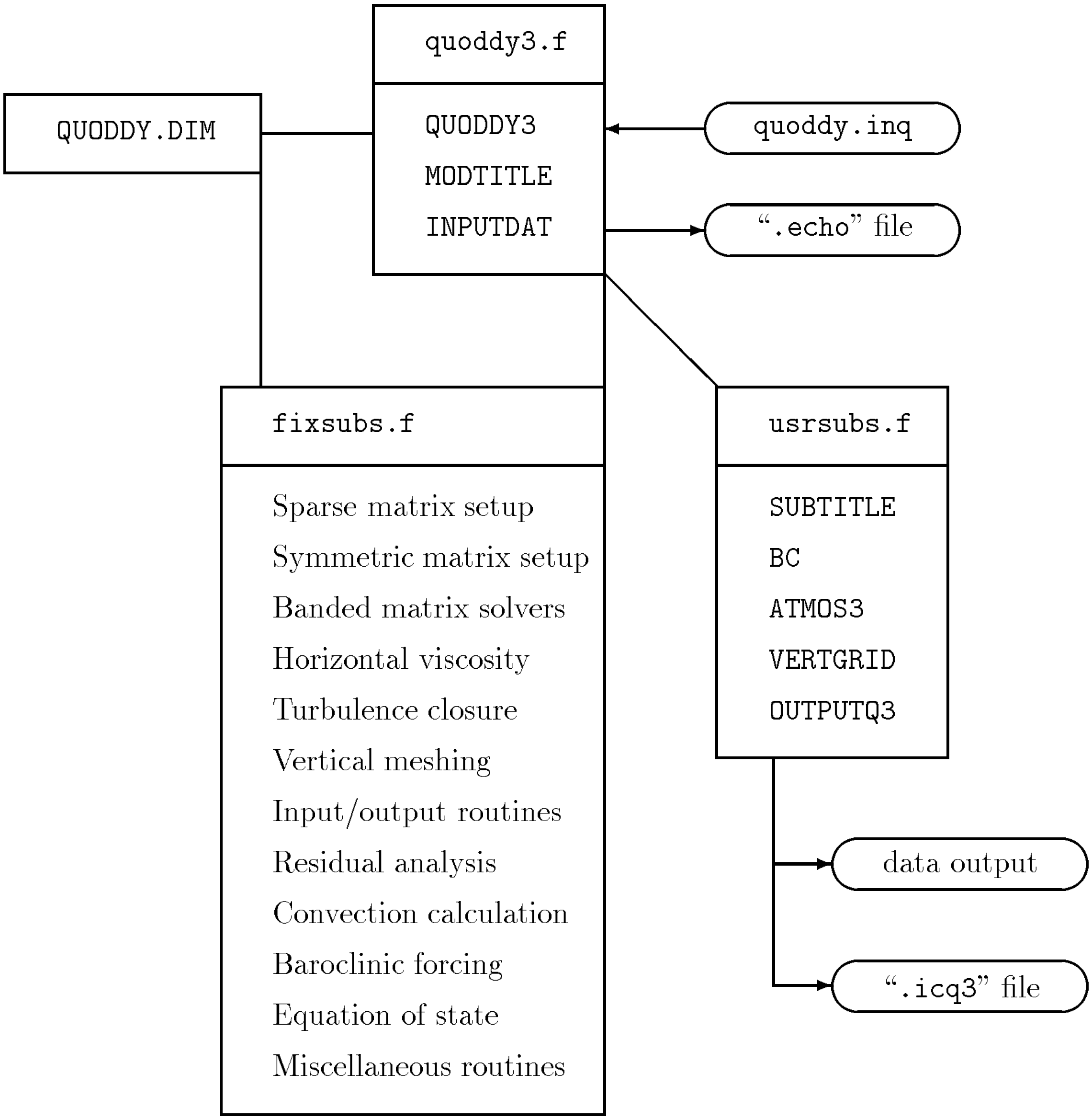

Figure 2: The general structure of the QUODDY3 software system.

3. *User subroutines*
   Each of the core programs must also be linked to a number of user-built subroutines in order to provide a record of the user's model, to specify the physical forcing, the vertical meshing, boundary conditions, and the manner in which results are to be written. The program `usrsubs.f` is a shell from which these subroutines may be constructed.

4. *Include file*
   The include file `QUODDY.DIM` assigns values to parameters required for variables dimensioning purposes. `QUODDY.DIM` is universal and is used in all the models in the QUODDY series.

In addition, the user interface for using the QUODDY series is the following:

1. *Creating an input file*
   Prior to running QUODDY1, QUODDY2, or QUODDY3, the user must first generate an input file, `quoddy.inq` that conforms to the "`.inq`" standard format (to be described in the next section). At run time, QUODDY will look for the input file, `quoddy.inq` for the respective simulation models. Any other user input/output is dependent on the user's treatment of the user subroutines.

2. *Creating the user subroutines*
   Five user subroutines — `SUBTITLE`, `BC`, `ATMOS3`, `VERTGRID`, `OUTPUTQ3` must also be linked to a QUODDY core program and `fixsubs.f`. Specifications for these subroutines are available in their respective shells, which contain the overall structure, argument list, and declarations sufficient as starting points for their construction.

3. *Analyzing the echo file*
   The echo file should be studied closely after each run with a new input file and/or a new set of user subroutines. The echo file contains useful information regarding the size of dimensioned arrays, boundary conditions, and run statistics.

3 Input-Output File Specifications

3.1 Input `quoddy.inq` file

The input file is a formatted `ASCII` data file conforming to the input requirements of QUODDY1, QUODDY2, and QUODDY3. The input file `quoddy.inq` contains file names of all of the mesh data required to run the QUODDY models and all the required input program variables. A sample file is presented in the Example Applications section.

In constructing the "`.inq`" input file, every line must start in the first column. The comments following one or more blank characters and beginning with either a "(" or a "`[`" character after the data input on each line are ignored except the first line of the file. Each input line of the `quoddy.inq` file is described as the following:

- Line 1 should read the character string, "{`Comment:`}", which is the label for the input data given in the next line and is ignored during data input.
- Line 2 inputs a one line comment of maximum 72 characters about the current simulation.
- Line 3 should read the character string, "{`Mesh name:`}", which is the label for the input data given in the next line and is ignored during data input.
- Line 4 inputs the mesh name, `MESH_NAME`, of the simulation of maximum 72 characters long with no blank character, and the input character string must start in the first column. With the `MESH_NAME` specified, it is assumed that there exist three mesh data files (symbolic linkage files can also be set up in UNIX environments) prefixed with `MESH_NAME` in the current directory. These required mesh data files must conform to the LABLIB standard (see Appendix) and they are the following:

  1. `MESH_NAME.nod` is the nodal Cartesian coordinates of the horizontal mesh. Each line is of the form

     `I    X(I)  Y(I)`

     where the data vectors `X(I)` and `Y(I)` are the x and y coordinates of the `I`th node, with the node index `I` running from 1 to the total number of nodes in the horizontal mesh.
  2. `MESH_NAME.ele` is the element incidence list of the linear triangles of the horizontal mesh. Each line is of the form

     `L    IN(1,L)  IN(2,L)  IN(3,L)`

     where the data array `IN(1:3,L)` is the element incidence list of the `L`th element, with the element index `L` running from 1 to the total number of elements in the mesh. It is important to notice that the first dimension of `IN` is the local elemental node number and the second dimension is the global element number.
  3. `MESH_NAME.bat` is the nodal bathymetric depth of the horizontal mesh. Each line contains

     `I    H(I)`

     where the data vector `H(I)` is the bathymetric depth of the `I`th node, with the node index `I` runs from 1 to the total number of nodes in the mesh.

  The total numbers of nodes and elements are determined by QUODDY during execution from reading and counting the data records in the `MESH_NAME.nod` and the `MESH_NAME.ele` files respectively.

- Line 5 should read the character string, "{`Boundary element incidence list:`}", which is the label for the input data given in the next line and is ignored during data input.

- Line 6 inputs the file name, `BEL_FILE`, of the boundary element incidence list of the horizontal mesh. `BEL_FILE` must be of maximum 72 characters long containing no blank character, and the input character string must start in the first column. The data file must conform to the LABLIB standard for the "`.bel`" file with the following format:
  - line 1 is the mesh name;
  - line 2 is a comment line;
  - line 3 and beyond are data lines containing the boundary element incidence list in the following format

    ```
    K    INBE(1,L)  INBE(2,L)  INBE(3,L)  INBE(4,L)
    ```

    where `K` is an integer index, the array entries `INBE(1,L)` and `INBE(2,L)` are the boundary element incidence list, `INBE(3,L)` and `INBE(4,L)` are the integer classification, `NTYPE`, of the left and right sides of the `L`th boundary element, while the index `L` runs from 1 to the total number of boundary elements in the horizontal mesh. Conventions for `NTYPE` are as follows:

0 :	interior
1 :	land
2 :	island
3 :	nonzero normal velocity
4 :	geostrophic outflow
5 :	surface elevation

  The total number of boundary elements is determined by QUODDY during execution from reading and counting the data records in the `BEL_FILE` file.

- Line 7 should read the character string, "{`Initial condition file:`}", which is the label for the input data given in the next line and is ignored during data input.
- Line 8 inputs the character string, `ICQ_FILE`, of maximum 72 characters long with no blank character, and the input character string must start in the first column. There are two possible entries of `ICQ_FILE` which govern the start of the simulation:
  - If the character string "`COLD-START`" is specified, the simulation is started at rest. All the field variables are initialized either to zero or to certain prescribed initial values and no initial condition file is needed.
  - If a file name with suffix "`.icq1`", "`.icq2`", or "`.icq3`" is specified, the simulation is a "hot start". On execution, QUODDY1, QUODDY2, or QUODDY3 will read the respective initial condition file, `ICQ_FILE` (a "`.icq1`", "`.icq2`", or "`.icq3`" file), and load the initial field variables according to the format specified by subroutine `ICQREAD`.
- Line 9 should read the character string, "{`Echo file:`}", which is the label for the input data given in the next line and is ignored during data input.
- Line 10 inputs the echo file name, `ECHO_FILE`, of maximum 72 characters long containing no blank character, and the input character string must start in the first column.
- Line 11 should read the character string, "{`Simulation parameters:`}", which is the general label for the input data given in the subsequent lines and this label line is ignored during data input.
- Line 12 inputs the dimensional units used in the simulation. For the QUODDY series, only SI units are allowed, therefore, line 6 should read "`SI UNITS`" starting from the first column. Otherwise, execution will halt and an error message will appear in the echo file.

- Line 13 inputs three entries of scaling factors for the x, y, and z coordinates respectively.
- Line 14 is the degrees latitude of the center of the mesh (e.g., 43.500 for the g2s mesh).
- Line 15 inputs the minimum depth. Nodal bathymetric depth will be altered to meet this minimum.
- Line 16 gives the starting time of the simulation in hours.
- Line 17 gives the length of the simulation in hours.
- Line 18 gives the time step increment in hours.
- Line 19 gives the maximum number of time steps allowed for the current simulation.
- Line 20 inputs two integers. The first is the number of time steps between calls to subroutine `OUTPUTQ1`, `OUTPUTQ2`, or `OUTPUTQ3`, and the second is the first time step when the output subroutine will be called.
- Line 21 inputs an integer key, `NONLIN`, of either 0 or 1, for switching off or on the nonlinear convective terms. For `NONLIN = 1`, the nonlinear convective terms are included in the calculation.
- Line 22 inputs the implicit gravity wave factor, `THETA`, between 0 and 1; `THETA = 1` for implicit, `THETA = 0` for explicit. `THETA = 0.75` is used for all the sample cases.
- Line 23 inputs the weighting factor, `TAU0 =` 2×10^{-4}`sec`, for the continuity equation in the shallow water wave equation.
- Line 24 is either the value of the spatially uniform horizontal viscosity (in MKS) or the Smagorinsky parameter used in the velocity dependent horizontal viscosity (a Smagorinsky parameter of 0.28 is used in [*Smagorinsky*, 1963]).
- Line 25 inputs the shoreline slip coefficient between 0 (free slip) and 1 (no slip).
- Line 26 inputs the total number of vertical nodes under each horizontal node.
- Line 27 inputs the vertical viscosity coefficient, N_{e0}, in the Davies-Furnes turbulence closure scheme, i.e., $N_e = N_{e0}\bar{v}^2$, where $\bar{v}$ is vertically averaged velocity. $N_{e0} = 0.2$ is used in [*Davies and Furnes*, 1980]. N_{e0} is not used in neither QUODDY2 nor QUODDY3.
- Line 28 inputs the dimensionless quadratic bottom stress drag coefficient, e.g., $C_d = 0.005$ is used in all the sample runs.
- Line 29 inputs the minimum value of the vertical viscosity at any point in time.
- Line 30 inputs the minimum value of the instantaneous bottom slip coefficient, $K_a = C_d v_{bot}$, where C_d is the dimensionless bottom stress drag coefficient, and v_{bot} is the speed at the bottom.
- Line 31 inputs an integer switch, `NLBS`, for choosing the nonlinear bottom stress: `NLBS = 0` for linear stress; `NLBS = 1` for quadratic stress.
- Line 32 inputs the time-weighting parameter, `EPSN` for vertical viscous stress: `EPSN = 1` for the fully implicit case; `EPSN = 0.5` for the Crank-Nicholson scheme.
- Line 33 inputs an integer key, `ISMAG`, of either 0 or 1, to signify the type of horizontal viscosity employed. For `ISMAG = 0`, uniform horizontal viscosity is used; for `ISMAG = 1`, Smagorinsky type velocity and grid size dependent horizontal viscosity is implemented.

3.2 Output "`.icq1`", "`.icq2`", and "`.icq3`", files for hot-start

The output files with suffix, "`.icq1`", "`.icq2`", and "`.icq3`" for hot-start are formatted `ACSII` data files conforming to the input requirements of the initial conditions for QUODDY1, QUODDY2, and QUODDY3 respectively. These hot-start files contain a snapshot of the current state of the simulation required to restart the simulation at the current time.

The exact format of these "`.icq`" hot-start files is as follows:

- Line 1 lists the model name. For the "`.icq1`" file, only the character string "`QUODDY1`", signifying the initial condition input for the QUODDY1 model, is allowed. Other character strings will halt the simulation. For the "`.icq2`" file, either "`QUODDY1`" or "`QUODDY2`" is allowed to accept either QUODDY1 or QUODDY2 type of initial condition state. For the "`.icq3`" file, "`QUODDY1`", "`QUODDY2`", or "`QUODDY3`" is allowed to accept any one of the three types of initial condition state.
- Line 2 lists the code name of the mesh, `MESH_NAME`, used in the simulation.
- Line 3 gives the input "`.inq`" file of the simulation which created the current hot-start file.
- Line 4 gives the core program which created the current hot-start file.
- Line 5 writes either `COLD-START` or the path of the initial condition file which created the current hot-start file.
- Line 6 inputs two integers. The first is the number of nodes in the horizontal mesh, and the second is the number of vertical nodes.
- Line 7 inputs an integer and two real numbers. First, the integer is the number of time steps the simulation has taken to create the current hot-start file. The two real numbers following are the current and the time step size (in hours) of this initial state.
- The next `NN` (the total number of nodes) lines give the vertically averaged free surface elevation and velocity at the current and the previous time step. Each line is of the form

  ```
  HMID(I)  UMID(I)  VMID(I)  HOLD(I)  UOLD(I)  VOLD(I)
  ```

  with nodal index `I` runs from 1 to `NN`.
- The next `NN`×`NNV` (`NNV` is the total number of vertical nodes) lines give the full 3-D description of all the field variables of the current state required for hot-start. The length of each line depends on the hot-start file employing.
  - The remaining "`.icq1`"data can be read by the following FORTRAN statement

    ```
    DO J=1,NNV
       DO I=1,NN
          READ (9,*) ZMID(I,J), ZOLD(I,J), UZMID(I,J), VZMID(I,J), WZMID(I,J)
       ENDDO
    ENDDO
    ```

    where `ZMID(I,J)` and `ZOLD(I,J)` are the z-coordinates at the current and the previous simulation times respectively, and `UZMID(I,J)`, `VZMID(I,J)`, and `WZMID(I,J)` are the three components of the 3-D velocity.
  - The remaining "`.icq2`"data can be read by the following FORTRAN statement

```
DO J=1,NNV
   DO I=1,NN
      READ (9,*) ZMID(I,J), ZOLD(I,J), UZMID(I,J), VZMID(I,J), WZMID(I,J),
                 Q2MID(I,J), Q2LMID(I,J), RHOMID(I,J)
   ENDDO
ENDDO
```

where `Q2MID(I,J)` and `Q2LMID(I,J)` are the 3-D turbulence variables, q^2 and $q^2 l$ with $q^2/2$ and l being the turbulence energy and the mixing length respectively, and `RHOMID(I,J)` is the 3-D density field at the current simulation time.

– The remaining "`.icq3`"data can be read by the following FORTRAN statement

```
DO J=1,NNV
   DO I=1,NN
      READ (9,*) ZMID(I,J), ZOLD(I,J), UZMID(I,J), VZMID(I,J), WZMID(I,J),
                 Q2MID(I,J), Q2LMID(I,J), TMPMID(I,J), SALMID(I,J)
   ENDDO
ENDDO
```

where `TMPMID(I,J)` and `SALMID(I,J)` are the 3-D temperature and salinity fields at the current simulation time respectively.

3.3 Output "`.nrv`" file

The output file with, "`.nrv`" is a formatted `ACSII` data file containing a snapshot of a vertical cut of a scalar field variable and velocity. Each "`.nrv`" file contains `NNV`×`NSEG` lines. Each line is of the form

```
I   ABS(I)  ORD(I)  SCAL(I)  VELN(I)  VELT(I)  VELZ(I)
```

where the data vectors, `ABS(I)` and `ORD(I)`, are the abscissa and ordinate of the coordinates of the defined nodes on the plane of the vertical cut, `SCAL(I)` is either the density, or temperature, or salinity, `VELN(I)`, `VELT(I)`, and `VELZ(I)` are the nodal normal, tangential, and vertical components of the velocity on the plane of the vertical cut. The index `I` runs from 1 to `NNV`×`NSEG`, which is the total number of nodes on the plane of the vertical cut, and `NNV` is the number of vertical nodes and `NSEG` is the number of defined horizontal nodes along the vertical cut.

4 Description of Programs

4.1 Core programs

There are three core programs available in the QUODDY series:

- `PROGRAM quoddy1.f`
- `PROGRAM quoddy2.f`
  There are version 1.0, `quoddy2.1.0.f`, version 1.1, `quoddy2.1.1.f`, version 1.2, `quoddy2.1.2.f`, and the most complete version 1.3, `quoddy2.1.3.f`, which should be the version to be used.
- `PROGRAM quoddy3.f`
  There are version 1, `quoddy3.1.f`, and version 2, `quoddy3.2.f` with the point source terms.

4.2 Fixed subroutines

The following are the eternally fixed subroutines used in the models in the QUODDY series:

- Sparse matrix manipulation routines:
  These routines are also available in the NML SPRSPAK [Lynch *et al.*, 1991].
  - `FUNCTION KAY`
  - `SUBROUTINE SPRSBLD1`
  - `SUBROUTINE SPRSBLD3DD`
  - `SUBROUTINE SPRSCONV`
  - `SUBROUTINE SPRSGRAD`
  - `SUBROUTINE SPRSINSRTD`
  - `SUBROUTINE SPRSINVMLT`
  - `SUBROUTINE SPRSMLT`
  - `SUBROUTINE SPRSMLTI`
  - `SUBROUTINE SPRSMLTIJ`
  - `SUBROUTINE SPRSMLTIN`
  - `SUBROUTINE SPRSROWZAP`
- Symmetric banded storage setup:
  - `SUBROUTINE RHSBUILD`
  - `SUBROUTINE RHSMULT`
- Banded matrix solvers:
  - `SUBROUTINE THOMAS`
  - `SUBROUTINE CTHOMAS`
  - `SUBROUTINE SOLVETR`
  - `SUBROUTINE BANSOLTR`
- Horizontal viscosity:

  - `SUBROUTINE SMAGOR`

- Turbulence closure schemes:
  - `SUBROUTINE DAVIES`
  - `SUBROUTINE QUADSTRESS`
  - `SUBROUTINE MY25`
  - `SUBROUTINE GALPERIN`
  - `SUBROUTINE GALPERIN2`
- Vertical meshing routines:
  - `SUBROUTINE UNISIGMA`
  - `SUBROUTINE SINEGRID`
  - `SUBROUTINE VERTGRD1`
  - `SUBROUTINE VERTGRD2`
- I/O and data analysis service subroutines:
  - `SUBROUTINE ICQREAD`
  - `SUBROUTINE ICQ1WRITE`
  - `SUBROUTINE ICQ2WRITE`
  - `SUBROUTINE ICQ3WRITE`
  - `SUBROUTINE RESIDUL1`
  - `SUBROUTINE RESIDUL2`
  - `SUBROUTINE VCUTSUB`
- Convection calculation:
  - `SUBROUTINE CONVECTION`
- Baroclinic pressure gradient:
  - `SUBROUTINE RHOXY`
  - `SUBROUTINE RHOXY2`
  - `SUBROUTINE RHOXYS`
  - `SUBROUTINE RHOXY2S`
- Equation of state:
  - `SUBROUTINE EQNSTATE0`
- Miscellaneous routines:
  - `SUBROUTINE HALFBW`
  - `SUBROUTINE INTTOCHR`
  - `SUBROUTINE VINTERPL`
  - `SUBROUTINE VINTERPL1`
  - `SUBROUTINE VINTERPL2`
  - `SUBROUTINE BASIS1D`

- `SUBROUTINE BASIS1D1`
- `SUBROUTINE ELEMCOEFS`
- `SUBROUTINE NODECOEFS`
- `SUBROUTINE VERTAVG`
- `SUBROUTINE VERTVEL`
- `SUBROUTINE VERTSUM`
- `SUBROUTINE VERTVEL3_2`

4.3 User subroutines

As previously mentioned, the appropriate user subroutines must be linked to the appropriate QUODDY core program and all the fixed subroutines in order to specified a simulation. The user subroutines are called to provide information about these subroutines to the echo file, and to specify the boundary conditions, the physical forcing, the vertical meshing, the equations of state, and the manner in which the results are to be written. Specifications for these subroutines are available in their respective shells, which contain the overall structure, argument list, dimensioning, and declarations sufficient as starting points for their construction.

4.3.1 Subroutine `SUBTITLE`

This subroutine must be modified by the user to provide summary information about all the user-defined subroutines. This information is output into the echo file. The argument list is

```
SUBROUTINE SUBTITLE (NFIL)
```

The only variable in the argument list is the logical number, `NFIL`, assigned to the echo file ("`.echo`") in the core program. In the subroutine SUBTITLE, comments about each of the user subroutines are given and are written to echo file of unit `NFIL` via the statements following the variable declaration statements. An empty subroutine can also be used if no information about the user subroutines is to be written to the echo file.

4.3.2 Subroutine `BC`

This subroutine must be modified by the user to assign proper barotropic boundary conditions. The argument list of subroutine BC is

```
SUBROUTINE BC (FFREQ, BC1, BC2, NNDIM, SSEC, ITER, III)
```

In the argument list, the variables,

`FFREQ`	is the prescribed tidal frequency of the forcing.
`BC1(1:NNDIM)`	is a data array of the set depth boundary values; or the x-velocity values of the set velocity boundary.
`BC2(1:NNDIM)`	is a data array of the y-velocity values of the set velocity boundary.
`NNDIM`	is the maximum number of horizontal nodes.
`SSEC`	is $t + \Delta t$ in seconds.

`ITER` is the corresponding time step number.

`III` is an integer flag for various types of boundary conditions to be considered.

There are four sections which require modification, beginning at statement labels 3, 4, 5, and 6. Each section starts with the appropriately labeled `CONTINUE` statement and ends with a `RETURN` statement.

1. *Forcing Frequency* (label 3)
   The first section assigns the frequency of the tidal forcing in radians per second to `FFREQ`. Time variation of the form $\exp(i\omega t)$ is assumed.
2. *Surface Elevation* (label 4)
   The proper free surface elevation boundary conditions are assigned in this section. A statement of the form

   ```
   BC1(I) = SVALUE
   ```

   must appear for each specified surface elevation node, where `I` is the horizontal node number and `SVALUE` is the free surface elevation for node `I`. It is noted that the main program initializes all required elevation boundary condition values to zero before subroutine BC is called. If the user does not specify the boundary condition for a node where one is required, the default of zero elevation is applied.
3. *Nonzero Normal Velocity* (label 5)
   Nonzero normal velocity boundary conditions are assigned in this section. A statement of the form

   ```
   BC1(I) = UVALUE
   ```

   must appear for each set nonzero normal velocity node, where `I` is the horizontal node number and `UVALUE` is the value of the normal velocity for node `I`. Positive normal velocity indicates flow out of the system. Note that boundary nodes where the normal velocity is zero are handled by default in QUODDY.
4. *Nonzero Velocity* (label 6)
   Nonzero velocity boundary conditions are assigned in this section. A statement of the form

   ```
   BC1(I) = XVALUE
   BC2(I) = VVALUE
   ```

   must appear for each set nonzero velocity node, where `I` is the horizontal node number and `XVALUE` and `YVALUE` are the values of the x and y-velocity for node `I`.

4.3.3 Subroutine `ATMOS3`

This subroutine must be modified by the user to assign proper values of atmospheric forcing terms, for wind stress and temperature flux at time `SSEC`. The argument list is

```
SUBROUTINE ATMOS3 (ATMX, ATMY, ATEMP, BTEMP, NN, SSEC, ITER)
```

In the argument list, the variables,

`ATMX(1:NNDIM)` is an array contains the x-component of the kinematic stress at the top of the water column.

`ATMY(1:NNDIM)` is an array contains the y-component of the kinematic stress at the top of the water column.

`ATEMP(1:NNDIM)` and `BTEMP(1:NNDIM)` are vector arrays containing the temperature flux, i.e., $N\frac{\partial T}{\partial z} = \texttt{BTEMP} - \texttt{ATEMP} * T$

`NN` is the number of horizontal nodes.

`SSEC` is $t + \Delta t$ in seconds.

`ITER` is the corresponding time step number.

Two statements of the following form must appear for each node where the atmospheric forcing is nonzero:

```
 ATMX(I) = XVALUE
 ATMY(I) = YVALUE
ATEMP(I) = AVALUE
BTEMP(I) = BVALUE
```

where `I` is the node number. The main program initializes all **ATMX** and **ATMY**, **ATEMP** and **BTEMP** values to zero before subroutine ATMOS3 is called. Therefore, the default of zero atmospheric forcing is applied if the user does not specify the **ATMX** and **ATMY**, **ATEMP** and **BTEMP** values for a node in subroutine ATMOS3.

4.3.4 Subroutine ATMOS3_2

This subroutine is very similar to ATMOS3 (described above) and is to be used with QUODDY3 version 2 only. The user must modify this subroutine to assign proper values of atmospheric forcing terms, for wind stress, temperature flux, and atmospheric source term, expressed as precipitation minus evaporation at time `SSEC`. The argument list is

```
SUBROUTINE ATMOS3_2 (ATMX, ATMY, ATEMP, BTEMP, PMEMID, NN, SSEC, ITER)
```

In the argument list, the variables are the same as in subroutine ATMOS3, and **PMEMID** is the atmospheric source terms, precipitation minus evaporation. The statement of the following form must appear for each node where the atmospheric source term is nonzero:

```
PMEMID(I) = PME_VALUE
```

where `I` is the node number. The main program initializes **PMEMID** values to zero before subroutine ATMOS3_2 is called. Therefore, the default of zero atmospheric source is applied if the user does not specify the **PMEMID** value for a node in subroutine ATMOS3_2.

4.3.5 Subroutine POINTSOURCE

As for subroutine ATMOS3_2, this subroutine is to be used with QUODDY3 version 2 only. This subroutine must be modified by the user to assign proper values of point source terms at time `SSEC`. The argument list is

```
SUBROUTINE POINTSOURCE (SRCMID, UZNEW, VZNEW, TNEW, SNEW, ZMID, NNDIM, NN, NEV,
                        SSEC, ITER)
```

`SRCMID(NNDIM,*)` are the point source terms. The main program initializes all `SRCMID` values to zero before subroutine POINTSOURCE is called. Therefore, the default of zero source term is applied if the user does not specify the `SRCMID` value for a node in subroutine POINTSOURCE.

NOTES relative to source terms:

- QUODDY versions prior to version 3.2 incorporate neither point sources nor net precipitation. Accordingly, subroutine POINTSOURCE is not needed and will be ignored; and the simpler subroutine ATMOS3 which lacks net precipitation in its argument list is invoked.
- QUODDY3.2 implicitly assumes that point sources of fluid are *neutral* relative to the turbulent variables q^2 and $q^2 l$ — i.e. the source fluid enters at the ambient in-situ value: $q^2_\sigma = q^2$ and $q^2 l_\sigma = q^2 l$. For that reason, POINTSOURCE accommodates no data for these variables.

4.3.6 Subroutine `VERTGRID`

This subroutine must be modified by the user to assign proper meshing in the vertical dimension. The argument list is

```
SUBROUTINE VERTGRID (NNDIM, NN, NNV, BATHY, HSURF, ZCORD)
```

In the argument list, the variables,

`NNDIM` is the maximum number of horizontal nodes.

`NN` is the total number of horizontal nodes.

`NNV` is the total number of vertical nodes.

`BATHY` is the bathymetric depth.

`HSURF` is the free surface elevation..

`ZCORD(1:NNDIM,1:NNV)` is a 3-D data array contains the vertical coordinates.

There are four vertical meshes subroutines provided which can be called in `VERTGRID` to provide different vertical gridding. These subroutines are

- `UNISIGMA` which defines a uniform ε-grid,
- `SINEGRID` which defines a ε-grid according to the sinusoidal transformation equation (22),
- `VERTGRD1` which defines a non-uniform ε-grid with at least 9 nodes uniformly in the top 100 m from the sea surface.
- `VERTGRD2` which defines another non-unfirom ε-grid with at least 9 nodes unfiromly in the top and bottom 100 m.

4.3.7 Subroutine `OUTPUTQ3`

This subroutine can be modified by the user to tailor the desired post-processing analysis and data output. Subroutine `OUTPUTQ3` is to be used with the QUODDY3 program. The argument list is

```
SUBROUTINE OUTPUTQ3 (CASNAM, FILINQ, FILICQ, FILSRC, ITER, HOURS, HMID, UMID, VMID,
                     ZMID, UZMID, VZMID, WZMID, Q2MID, Q2LMID, RHOMID, TMPMID,
                     SALMID, DELTHR, HOLD, UOLD, VOLD, ZOLD, HDOWN, NE, NN, NNDIM,
                     NNV, AHI, XCORD, YCORD, INC, ENO, CD, ENZM, ENZH, ENZQ)
```

with the following declaration statements

```
INTEGER ITER, NE, NN, NNDIM, NNV, INC(3,*)
CHARACTER*(*) CASNAM
CHARACTER*72 FILICQ, FILINQ, FILSRC
REAL DELTHR, HOURS, ENO, CD
REAL AHI(*), HMID(*), UMID(*), VMID(*), HOLD(*), UOLD(*), VOLD(*), HDOWN(*),
     ZMID(NNDIM,*), ZOLD(NNDIM,*), UZMID(NNDIM,*), VZMID(NNDIM,*), WZMID(NNDIM,*),
     Q2MID(NNDIM,*), Q2LMID(NNDIM,*), RHOMID(NNDIM,*), SALMID(NNDIM,*),
     TMPMID(NNDIM,*), ENZM(NNDIM,*), ENZH(NNDIM,*), ENZQ(NNDIM,*), XCORD(*),
     YCORD(*)
```

This subroutine is responsible for writing the simulation data output to the user's specification. The `OUTPUTQ3` shell contains specifications and array dimensioning for the arrays, and some useful post-processing routines. Similarly, subroutines `OUTPUTQ1` and `OUTPUTQ2` are to be used with the QUODDY1 and QUODDY2 models respectively with very similar argument lists and structures.

5 General Notes

The following points on the general program structure and details are worth noticing for users in using QUODDY1, QUODDY2, and QUODDY3. They are

1. The triangular horizontal grid node numbering system is retained intact by use of a double-subscript node numbering convention. Node `(I,J)` indicates horizontal position `I`, vertical position `J`. All 3-D arrays use this convention. For example, `U(I,J)` indicates the x-component of velocity at horizontal node `I` and vertical node `J`.

2. Recall that there are the same number of vertical nodes everywhere with the same spacing on the ϵ-coordinate, however, the bathymetric depth may be different at different nodes. Therefore equal values of `J` do not necessarily imply equal values of either z or z/h.

3. The vertical coordinate z and the vertical node numbering are positive upward:

   $$\begin{array}{rll} \text{Bottom:} & \texttt{J=1}, & z = -h \\ \text{Top:} & \texttt{J=NNV}, & z = \zeta \end{array}$$

   where `NNV` is the total number of vertical nodes, h is the bathymetric depth, and ζ is the free surface elevation.

4. Consistent with the slip condition at the bottom, the computational bottom $z_b = -h$ is located within the constant stress layer at a height $\xi_b = 1$ m above the actual sea floor.

5. QUODDY1, QUODDY2, and QUODDY3 are coded in FORTRAN with single precision data types for all the `COMPLEX` and `REAL` variables. These data types must be respected in user-defined subroutines. The declarations provided in the user subroutine shells are complete and unambiguous. Extra care must be exercised relative to mixed-mode computations.

6. SI (MKS) units are used in the program. All physical quantities retain their original dimensions. The only exceptions relate to density and density-related arrays:

   - Subroutine EQSTATE0 reports $\rho - 1000$ in MKS unit for precision reasons.
   - The internal convention for density, which is carried internally in QUODDY as the normalized density anomaly $(\rho - \rho_{ref})/\rho_{ref}$.
   - An additional detail is that density is by definition represented in σ_t units in all `.lst` files.

7. The baroclinic pressure gradient is computed utilizing level surfaces as in Naimie *et al.* [1994]. The density is interpolated from the FEM vertical grid onto level triangular grids at fixed depths (10 m spacing from surface to 60 m; 15 m spacing to 75 m; 25 m spacing to 250 m; 50 m spacing to 500 m; and 100 m below). $\nabla_{xy}\rho$ is computed on these surfaces and interpolated back to the original vertical grid, where the appropriate vertical integrations are performed.

8. QUODDY versions prior to version 3.2 incorporate neither point sources nor net precipitation. Accordingly, subroutine POINTSOURCE is not needed and will be ignored; and the simpler subroutine ATMOS3 which lacks net precipitation in its argument list is invoked.

9. QUODDY3.2 implicitly assumes that point sources of fluid are *neutral* relative to the turbulent variables q^2 and $q^2 l$ — i.e. the source fluid enters at the ambient in-situ value: $q_\sigma^2 = q^2$ and $q^2 l_\sigma = q^2 l$. For that reason, POINTSOURCE accommodates no data for these variables.

6 Example Applications

As an illustration of the usage of the QUODDY software, four sample simulations of application to the Georges Bank in the Gulf of Maine Region are presented. The 2-D horizontal mesh employed is the g2s mesh used in Lynch *et al.* [1995]. All the simulations were conducted on an IBM RS/6000 model 355 workstation using the QUODDY2 version 1.3 software.

In general, the simulation protocol involves the following steps:

1. Create a directory to host the simulation.
2. Copy all the necessary QUODDY source code, fixed subroutines, user subroutines, include file for dimensioning, and data files such as the geometry files, boundary file for tidal forcing, initial condition file (not needed for cold-start runs), and the input file (`quoddy.inq`) to the host directory.
3. Comply and link the QUODDY source code with all the fixed and user subroutines.
4. Edit the input file, `quoddy.inq` to set up the desired simulation.
5. Run the simulation and examine the output files.

All the files required for the simulations are described and the paths of all the related input/output data files are listed in detail below for each case. The essential points of the simulation are also highlighted. The simulation results are not shown here, but can be found in Lynch *et al.* [1995].

6.1 Barotropic Tidal Rectification

6.1.1 Example 1

- *Description*
  As a starting point, a simulation of cold-start from fluid at rest was carried out under constant density and was forced with M_2 tides as in Lynch and Naimie [1993]. The interest was in the response after one M_2 tidal period (12.42 hours).
- *Source programs*
  Main program — `quoddy2.1.3.f`
  Fixed subroutines — `fixsubs.f`
  User subroutines — `usrsubs.f`
  Include file — `QUODDY.DIM`
- *Input data files*
  Geometry files — `g2s.nod` (nodal coordinates of the horizontal mesh), `g2s.ele` (incidence list), `g2s.bat` (nodal bathymetry)
  Boundary file — `g2sm2.bel` (boundary element incidence list)
  Input file — `quoddy.inq`
- *Output files*
  Echo file — `g2sm2.echo`
  Eulerian velocity files — `g2s1a.v2r` (depth averaged velocity and averaged over a M_2 tidal period)
  Surface elevation file — `g2s1.s2r` (averaged over one M_2 tidal period)
  Hot-start file — `RESTART.icq2`
- *File locations*
  The above mentioned files are located in the GLOBEC directory,
  `/usr/habitat/GLOBEC/quoddy2/example1`

- *Simulation notes*
  The include file, `QUODDY.DIM` for the g2s mesh should include the following declaration statements,

```
NNDIM, NEDIM, NBWE, NFTRDIM, NBEDIM, NNVDIM, NRDIM, NB1DIM
PARAMETER (NNDIM=6756, NEDIM=12877, NBWE=305, NFTRDIM=20, NBEDIM=1000,
          NNVDIM=21, NRDIM=100, NB1DIM=NBWE/2+1)
```

  The input file, `quoddy.inq` for this simulation is the following,

```
{Comment:}
GoM - g2s w/ M2 forcing, quoddy2 version 1.3, barotropic tide
{Mesh name:}
g2s
{Boundary element incidence list:}
g2sm2.bel
{Initial condition file:}
COLD-START
{Echo file:}
barotropic.echo
{Simulation parameters:}
SI UNITS                [units]
1.00   1.00   1.00      [x, y, and z scaling factor]
43.50                   [degree latitude]
10.00                   [minimum depth]
0.00                    [starting time (hours)]
298.08                  [length of simulation (hrs.)]
0.048515625298          [time step (hours)]
256                     [maximum number of time steps]
1        1              [time steps between and before output]
1                       [convection factor, NONLIN]
0.75                    [THETA]
0.0002                  [TAU0]
0.28                    [horizontal viscosity or Smagorinsky parameter]
0.00                    [sidewall slip coefficient]
21                      [number of vertical nodes]
0.20                    [vertical viscosity coefficient (Davies-Furnes)]
0.0050                  [quadratic drag coefficient]
0.0020                  [minimum viscosity]
0.00001                 [minimum bottom stress]
1                       [nonlinearity factor, NLBS]
0.50                    [implicity factor, EPSN]
1                       [horizontal viscosity type, ISMAG]
```

6.1.2 Example 2

- *Description*
  As a second example, a simulation of hot-start from fluid with an initial state of a well established barotropic tidal rectification simulation, was carried out identical to Example 1. Again the interest was in the response after one M_2 tidal cycle (12.42 hours) after the hot-start.

- *Source programs*
  Main program — `quoddy2.1.3.f`
  Fixed subroutines — `fixsubs.f`
  User subroutines — `usrsubs.f`
  Include file — `QUODDY.DIM`

- *Input data files*
  Geometry files — `g2s.nod`, `g2s.ele`, `g2s.bat`
  Boundary file — `g2sm2.bel`
  Input file — `quoddy.inq`
  Initial condition — `g2shot_21.icq2`

- *Output files*
  Echo file — `barotide.echo`
  Eulerian velocity files — `g2s1a.v2r`
  Surface elevation file — `g2s1.s2r`
  Hot-start file — `RESTART.icq2`

- *File locations*
  The above mentioned files are located in the GLOBEC directory,
  `/usr/habitat/GLOBEC/quoddy2/example2`

- *Simulation notes*
  The include file, `QUODDY.DIM` for the g2s mesh is the same as in Example 1. The input file, `quoddy.inq` for this simulation is the same as in Example 1 except the 8th, the 10th, and the 16th line should read,

```
g2shot.icq2
barotide.echo
298.08          [starting time (hours)]
```

  respectively instead.

6.2 Wind-Driven Mixed Layer

6.2.1 Example 3

- *Description*
  As a third example, a simulation identical to Example 2, in addition, forced with a constant wind stress of 0.1 Pascal directed towards 118.5 degrees clockwise from true north with an insulated sea surface, and 41 vertical nodes were employed instead of 21. We are interested in the response after one M_2 tidal cycle (12.42 hours) after hot-start.

- *Source programs*
  Main program — `quoddy2.1.3.f`
  Fixed subroutines — `fixsubs.f`
  User subroutines — `usrsubs.f`
  Include file — `QUODDY.DIM`

- *Input data files*
  Geometry files — `g2s.nod`, `g2s.ele`, `g2s.bat`
  Boundary file — `g2sm2.bel`
  Input file — `quoddy.inq`
  Initial condition — `g2shot_41.icq2`

- *Output files*
  Echo file — `barowind.echo`
  Time-series files — `g2s32.dat`, ..., `g2s256.dat` are eight M_2 tidal time result files over the Wilkinson Basin and the Jordan Basin containing turbulent kinetic energy and vertical viscosity data. Each line has the following format:

  ```
  I    Z  KE_J  Nm_J  KE_W  Nm_W
  ```

  where `I` is the vertical node number, `Z` is the vertical coordinate, `KE_J` and `Nm_J` are the turbulent kinetic energy and vertical viscosity at a node over the Jordan Basin, `KE_W` and `Nm_W` are the same quantities over the Wilkinson Basin.

- *File locations*
  The above mentioned files are located in the GLOBEC directory,
  `/usr/habitat/GLOBEC/quoddy2/example3`

- *Simulation notes*
  The include file, `QUODDY.DIM` for the g2s mesh is the same as in Example 1, except `NNVDIM=41`. The input file, `quoddy.inq` for this simulation is the same as in Example 1 except the 10th and the 26th line should read `barowind.echo` and `41` respectively instead. The following statements are needed (with proper dimensioning of the variables) in subroutine ATMOS2 among the user provided subroutines,

  ```
  PI = ACOS(-1.)
  DENSITY = 1000.
  PASCAL = 0.0472
  ANGLE = 121.4
  GEOFAC = ANGLE*PI/180.
  DO I=1,NN
     ATMX(I) = PASCAL*SIN(GEOFAC)/DENSITY
     ATMY(I) = PASCAL*COS(GEOFAC)/DENSITY
  ENDDO
  ```

6.2.2 Example 4

- *Description*
  As a forth and final illustration, a simulation identical to Example 3 was set up with an initially stratified density field and was allowed to evolve in tidal time. We are interested in the response after one M_2 tidal cycle (12.42 hours) after the hot-start.

- *Source programs*
  Main program — `quoddy2.1.3.f`
  Fixed subroutines — `fixsubs.f`
  User subroutines — `usrsubs.f`
  Include file — `QUODDY.DIM`

- *Input data files*
  Geometry files — `g2s.nod`, `g2s.ele`, `g2s.bat`
  Boundary file — `g2sm2.bel`
  Input file — `quoddy.inq`
  Initial condition — `g2shot_41.icq2`

- *Output files*
  Echo file — `strtwind.echo`
  Time-series files — `g2s32.dat`, ... `g2s256.dat` are eight M_2 tidal time result files over the Wilkinson Basin containing turbulent kinetic energy, vertical viscosity, and density data. Each line has the following format:

  ```
  I    Z  Q2  ENZM  RHO
  ```

  where `I` is the vertical node number, `Z` is the vertical coordinate, `Q2`, `ENZM`, and `RHO` are the turbulent kinetic energy, vertical eddy viscosity, and density at a node over the Wilkinson Basin.

- *File locations*
  The above mentioned files are located in the GLOBEC directory,
  `/usr/habitat/GLOBEC/quoddy2/example4`

- *Simulation notes*
  The initial stratification was set up as initial condition using the following statement,

  ```
  STRAF = 0.0005
  DO I=1,NN
     DO J=1,NNV
        RHOMID(I,J) = -STRAF*ZMID(I,J)/100.
        IF (ZMID(I,J) .GT. 100.)  RHOMID(I,J) = -STRAF
     ENDDO
  ENDDO
  ```

  The include file, `QUODDY.DIM` for the g2s mesh is the same as in Example 1. The input file, `quoddy.inq` for this simulation is the same as in Example 1 except the 10th line should read `strtwind.echo` instead.

7 Appendix: QUODDY3.2 Source Terms

In QUODDY3.2, sources are introduced in the governing equations to provide for mass entry at any point in the system, including along no-flux boundaries. The essential changes require addition of source notation, the alteration of the primitive equations, and the systematic projection of these alterations throughout the equations which form the actual discrete system. The complete equation set is presented in the previous appendix; here we list the essential source-term modifications to the primitive equations. These source effects are created in the user-defined subroutine POINTSOURCE. The conditions at the free surface are also summarized here; they are created in the user-defined subroutines ATMOS3 or ATMOS3_2.

Point Source Notation:

$\sigma(x,y,z,t)$ is a distributed mass source (mass/time/unit volume);
σ/ρ is the volumetric source (volume/time/unit volume);
$\mathbf{v}_\sigma$, T_σ, S_σ, q^2_σ, q^2l_σ are the properties of the source fluid.

Free Surface Source Notation:

$(P-E)$ is (Precipitation - Evaporation) at the free surface: volume/time/unit area;
A and B characterize the Type III heat flux condition: $N\frac{\partial T}{\partial z} = A - BT$

Continuity:

$$\frac{1}{\rho}\frac{d\rho}{dt} + \nabla\cdot\mathbf{v} = \frac{\sigma}{\rho} \tag{A0}$$

Momentum:

$$\frac{d\mathbf{v}}{dt} + \ldots = \frac{\sigma}{\rho}(\mathbf{v}_\sigma - \mathbf{v}) \tag{A1}$$

Conservation:

$$\frac{dT}{dt} + \ldots = \frac{\sigma}{\rho}(T_\sigma - T) \tag{A2}$$

$$\frac{dS}{dt} + \ldots = \frac{\sigma}{\rho}(S_\sigma - S) \tag{A3}$$

$$\frac{dq^2}{dt} + \ldots = \frac{\sigma}{\rho}(q^2_\sigma - q^2) \tag{A4}$$

$$\frac{dq^2l}{dt} + \ldots = \frac{\sigma}{\rho}(q^2l_\sigma - q^2l) \tag{A5}$$

Free Surface:

$$\frac{\partial\zeta}{\partial t} + \mathbf{v}\cdot\nabla_{xy}\zeta = W + (P-E) \tag{A6}$$

$$N\frac{\partial T}{\partial z} = A - BT \tag{A7}$$

Vertically integrated continuity:

$$\frac{\partial\zeta}{\partial t} + \nabla_{xy}\cdot\int_{-h}^{\zeta}\mathbf{v}dz = \int_{-h}^{\zeta}\frac{\sigma}{\rho}dz + (P-E) \tag{A8}$$

Wave Equation:

$$\left(\frac{\partial}{\partial t} + \tau_0\right)\frac{\partial\zeta}{\partial t} + \nabla_{xy}\cdot\left(\frac{\partial}{\partial t} + \tau_0\right)\int_{-h}^{\zeta}\mathbf{v}dz = \left(\frac{\partial}{\partial t} + \tau_0\right)\left[\int_{-h}^{\zeta}\frac{\sigma}{\rho}dz + (P-E)\right] \tag{A9}$$

NOTES relative to source terms:

- QUODDY versions prior to version 3.2 incorporate neither point sources nor net precipitation. Accordingly, subroutine POINTSOURCE is not needed and will be ignored; and the simpler subroutine ATMOS3 which lacks net precipitation in its argument list is invoked.
- QUODDY3.2 implicitly assumes that point sources of fluid are *neutral* relative to the turbulent variables q^2 and q^2l — i.e. the source fluid enters at the ambient in-situ value: $q^2_\sigma = q^2$ and $q^2l_\sigma = q^2l$. For that reason, POINTSOURCE accommodates no data for these variables.

8 Appendix: Model Equations and Numerical Methods

We solve the 3-D hydrodynamic equations with the conventional Boussinesq and hydrostatic assumptions. Temperature and salinity are transported, and density is determined via an equation of state. The subgrid-scale dissipation is represented in eddy viscosity (diffusivity) form. This is parameterized in terms of stratification plus turbulent kinetic energy and mixing length, both of which evolve at the macroscale.

8.1 Notation

The following notation is used:

$\mathbf{v}(x, y, z, t)$ is the fluid velocity, with Cartesian components (u, v, w),
$\bar{\mathbf{v}}(x, y, t)$ is the vertical average of $\mathbf{v}$,
$\zeta(x, y, t)$ is the free surface elevation,
$h(x, y)$ is the bathymetric depth (more precisely, the depth of a position in the bottom constant-stress layer at which boundary conditions are applied, typically about 1 m above the sea floor,
$H(x, y, t)$ is the total fluid depth, $H = h + \zeta$,
$\rho(x, y, z, t)$ is the fluid density; ρ_0 is a reference value,
$T(x, y, z, t)$ is the fluid temperature,
$S(x, y, z, t)$ is the fluid salinity,
$q^2(x, y, z, t)/2$ is the turbulent kinetic energy,
$l(x, y, z, t)$ is the turbulent mixing length,
$N_m(x, y, z, t)$ is the vertical eddy viscosity,
$N_h(x, y, z, t)$ is the vertical eddy diffusivity for heat and salt,
$N_q(x, y, z, t)$ is the vertical eddy diffusivity for q^2 and $q^2 l$,
$\mathbf{F}_m, F_T, F_S$ are non-advective horizontal exchanges of momentum, heat, and salt,
g is gravity,
$\mathbf{f}$ is the Coriolis vector, directed vertically with magnitude f
∇ is the gradient operator; ∇_{xy} is its horizontal part,
$\frac{d}{dt}$ is the material derivative following the fluid motion in 3-D, $\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla$
(x, y) are the horizontal Cartesian coordinates,
z is the vertical coordinate, positive upward; $-h \leq z \leq \zeta$,
t is time.

Source Notation:

$\sigma(x, y, z, t)$ is a distributed mass source (mass/time/unit volume);
σ/ρ is the volumetric source (volume/time/unit volume);
$\mathbf{v}_\sigma$, T_σ, S_σ, q^2_σ, $q^2 l_\sigma$ are the properties of the source fluid;
P is Precipitation at the free surface: volume/time/unit area;
E is Evaporation at the free surface: volume/time/unit area.

8.2 Governing Equations

There are six canonical 3-D state variables for which conventional transport equations are written. We have the two horizontal components of the momentum equations:

$$\frac{d\mathbf{v}}{dt} + \mathbf{f} \times \mathbf{v} + g\nabla_{xy}\zeta - \frac{\partial}{\partial z}\left(N_m \frac{\partial \mathbf{v}}{\partial z}\right) = -\frac{g}{\rho_0}\int_z^\zeta \nabla_{xy}\rho dz + \mathbf{F}_m + \frac{\sigma}{\rho}(\mathbf{v}_\sigma - \mathbf{v}) \tag{1}$$

heat and salt conservation:

$$\frac{dT}{dt} - \frac{\partial}{\partial z}\left(N_h \frac{\partial T}{\partial z}\right) = F_T + \frac{\sigma}{\rho}(T_\sigma - T) \tag{2}$$

$$\frac{dS}{dt} - \frac{\partial}{\partial z}\left(N_h \frac{\partial T}{\partial z}\right) = F_S + \frac{\sigma}{\rho}(S_\sigma - S) \tag{3}$$

and equations for the turbulent kinetic energy and mixing length:

$$\frac{dq^2}{dt} - \frac{\partial}{\partial z}\left(N_q \frac{\partial q^2}{\partial z}\right) = 2\left[N_m\left(\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2\right) + \frac{g}{\rho_0}N_h\frac{\partial \rho}{\partial z}\right] - 2\left[\frac{q^3}{B_1 l}\right] + \frac{\sigma}{\rho}(q_\sigma^2 - q^2) \tag{4}$$

$$\frac{dq^2 l}{dt} - \frac{\partial}{\partial z}\left(N_q \frac{\partial q^2 l}{\partial z}\right) = lE_1\left[N_m\left(\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2\right) + \frac{g}{\rho_0}N_h\frac{\partial \rho}{\partial z}\right] - lW\left[\frac{q^3}{B_1 l}\right] + \frac{\sigma}{\rho}(q^2 l_\sigma - q^2 l) \tag{5}$$

wherein E_1 and B_1 are experimental constants [Mellor and Yamada, 1982] and W is a wall proximity function [Blumberg *et al.*, 1992] (Table 1). The final state variable is the free surface $\zeta(x, y, t)$ whose evolution is defined by the vertically integrated continuity equation

$$\frac{\partial \zeta}{\partial t} + \nabla_{xy} \cdot \int_{-h}^{\zeta} \mathbf{v} dz = \int_{-h}^{\zeta} \frac{\sigma}{\rho} dz + (P - E) \tag{6}$$

The system is closed with several equilibrium relations. The 3-D continuity equation provides a means for computing vertical velocity w in terms of the horizontal velocity:

$$\frac{\partial w}{\partial z} = -\nabla_{xy} \cdot \mathbf{v} + \frac{\partial}{\partial z}(\frac{\sigma}{\rho}) \tag{7}$$

The density is related to temperature and salinity by the equation of state [Gill, 1982]:

$$\rho = \rho(T, S) \tag{8}$$

evaluated at constant pressure. The closure for the vertical turbulent mixing coefficients is

$$N_m = qls_m \tag{9a}$$

$$N_h = qls_h \tag{9b}$$

$$N_q = qls_q \tag{9c}$$

where s_q is a constant and the stability functions s_m and s_h are simple algebraic functions of the local stratification $G_h \equiv \frac{l^2}{q^2}\frac{g}{\rho_0}\frac{\partial \rho}{\partial z}$ [Galperin *et al.* 1988]:

Table 1: Summary of constants associated with the vertical closure. Column I contains the Mellor-Yamada [1982] experimental constants, plus E_3 from Blumberg *et al.* [1992] and von Karman's constant κ. Column II contains the formulas from Galperin *et al.* [1988] used in equations (9) herein. Their numerical values are recorded in Column III.

I	II	III
$A_1 = .92$	$g_0 \equiv 1. - 6A_1/B_1$	$g_0 = .66747$
$A_2 = .74$	$g_1 \equiv 6A_1 + B_2$	$g_1 = 15.620$
$B_1 = 16.6$	$g_2 \equiv A_1(g_0 - 3C_1)$	$g_2 = .39327$
$B_2 = 10.1$	$g_3 \equiv 3A_1A_2[(B_2 - 3A_2)g_0 - 3C_1g_1]$	$g_3 = 3.0858$
$C_1 = .08$	$g_4 \equiv 3A_2g_1$	$g_4 = 34.676$
$E_1 = 1.8$	$g_5 \equiv 9A_1A_2$	$g_5 = 6.1272$
$E_2 = 1.33$	$g_6 \equiv A_2g_0$	$g_6 = .49393$
$E_3 = 0.25$		
$\kappa = 0.4$		

$$s_m = \frac{(g_2 - g_3G_h)}{(1. - g_4G_h)(1. - g_5G_h)} \tag{9d}$$

$$s_h = \frac{g_6}{(1. - g_4G_h)} \tag{9e}$$

$$s_q = 0.2 \tag{9f}$$

The constants g_i, introduced here for simplicity, are related in Table 1 to the standard constants of Mellor and Yamada [1982] derived from experiment. Also from Galperin *et al.* [1988] we enforce an upper bound on the mixing length under stable stratification:

$$l \leq \frac{0.53q}{\sqrt{-\frac{g}{\rho_0}\frac{\partial\rho}{\partial z}}} \tag{10}$$

This implies an effective lower limit $G_h \geq -0.28$. Under unstable stratification, G_h is capped at the upper limit $G_h \leq 0.0233$. Among other things, this prevents the computation of negative diffusivity according to (9d) and (9e). Finally, the non-advective horizontal exchanges $\mathbf{F}_m$, F_T, etc. are all expressed in Laplacian forms, e.g.

$$\mathbf{F}_m = \nabla_{xy} \cdot (A\nabla_{xy}\mathbf{v}) \tag{11}$$

with A given in terms of the local shear and grid scale δ as in Smagorinsky [1963]:

$$A = 0.28\delta^2\sqrt{\left(\frac{\partial\bar{u}}{\partial x} - \frac{\partial\bar{v}}{\partial y}\right)^2 + \left(\frac{\partial\bar{v}}{\partial x} + \frac{\partial\bar{u}}{\partial y}\right)^2} \tag{12}$$

8.3 Boundary Conditions

We solve these equations subject to conventional horizontal boundary conditions as described in previous studies. Vertical boundary conditions are posed as follows. For the horizontal velocity, the atmospheric shear stress is specified at the surface

$$N_m \frac{\partial \mathbf{v}}{\partial z}\bigg|_{z=\zeta} = H\mathbf{\Psi} \tag{13a}$$

At the bottom we use a conventional quadratic slip condition relating shear stress to the bottom velocity $\mathbf{v}_b$ via the dimensionless quadratic bottom stress drag coefficient, C_d

$$N_m \frac{\partial \mathbf{v}}{\partial z}\bigg|_{z=-h} = C_d|\mathbf{v}_b|\mathbf{v}_b \tag{13b}$$

Atmospheric heat input is specified at the surface as a "Type III" or radiation condition with heating rate α and equilibrium temperature T_0:

$$N_h \frac{\partial T}{\partial z}\bigg|_{z=\zeta} = -\alpha(T - T_0) \tag{14a}$$

At the bottom, heat flux is assumed negligible

$$N_h \frac{\partial T}{\partial z}\bigg|_{z=-h} = 0 \tag{14b}$$

Analogous no-flux conditions are imposed on S at surface and bottom. For q^2, Dirichlet conditions are enforced at the bottom:

$$q^2 = B_1^{2/3} u_*^2 \tag{15}$$

with the friction velocity $u_*^2 = |N_m \frac{\partial \mathbf{v}}{\partial z}|$ specified as in the velocity boundary conditions. Consistent with the slip condition at the bottom, the computational bottom $z_b = -h$ is located within the constant stress layer at a height $\xi_b = 1$ m above the actual sea floor. The mixing length at the computational bottom is set in accord with the law of the wall:

$$l = \kappa \xi_b \tag{16}$$

where $\kappa = 0.4$ is von Karman's constant. At the free surface, no-flux (Neuman) conditions are applied to both q^2 and $q^2 l$. The wall proximity function W is taken from Blumberg *et al.* [1992] and incorporates asymmetry between surface and bottom "walls":

$$W = 1 + E_2\left[\frac{l}{\kappa(z - z_b + \xi_b)}\right]^2 + E_3\left[\frac{l}{\kappa(\zeta - z + \xi_s)}\right]^2 \tag{17}$$

with $l/\kappa d$ approaching unity as the distance d from either wall vanishes. Both ξ_b and ξ_s are set equal to 1. Finally, kinematic conditions are enforced on the vertical velocity w at the surface,

$$w = \frac{\partial \zeta}{\partial t} + \mathbf{v} \cdot \nabla_{xy}\zeta - (P - E) \tag{18a}$$

and at the bottom,

$$w = -\mathbf{v} \cdot \nabla_{xy} h \tag{18b}$$

Both are enforced as Dirichlet conditions on the z-derivative of (7), as in Lynch and Naimie [1993].

8.4 Wave Continuity Equation

For the sea surface elevation ζ, we rearrange (6) and the vertically integrated momentum equations to obtain the Shallow Water Wave Equation as in Lynch and Werner [1991]:

$$\begin{aligned}\frac{\partial^2\zeta}{\partial t^2} + \tau_0\frac{\partial\zeta}{\partial t} - \nabla_{xy}\cdot\Bigg[-\mathbf{v}|_{z=\zeta}\frac{\partial\zeta}{\partial t} + \int_{-h}^{\zeta}\left[\mathbf{v}\cdot\nabla\mathbf{v} + \frac{g}{\rho_0}\int_z^{\zeta}\nabla_{xy}\rho dz' - \mathbf{F}_m - \frac{\sigma}{\rho}(\mathbf{v}_\sigma - \mathbf{v})\right] dz \\ +gH\nabla_{xy}\zeta + \mathbf{f}\times H\bar{\mathbf{v}} - \tau_0 H\bar{\mathbf{v}} - N_m\frac{\partial\mathbf{v}}{\partial z}\bigg|_{z=\zeta} + N_m\frac{\partial\mathbf{v}}{\partial z}\bigg|_{z=-h}\Bigg] = \left(\frac{\partial}{\partial t} + \tau_0\right)\left[\int_{-h}^{\zeta}\frac{\sigma}{\rho}dz + (P-E)\right]\end{aligned} \tag{19}$$

where τ_0 is a numerical constant [Kinnmark 1986]. Inserting the boundary conditions (13) into (19), we arrive at the final form of the Wave Equation:

$$\begin{aligned}\frac{\partial^2\zeta}{\partial t^2} + \tau_0\frac{\partial\zeta}{\partial t} - \nabla_{xy}\cdot\Bigg[-\mathbf{v}|_{z=\zeta}\frac{\partial\zeta}{\partial t} + \int_{-h}^{\zeta}\left[\mathbf{v}\cdot\nabla\mathbf{v} + \frac{g}{\rho_0}\int_z^{\zeta}\nabla_{xy}\rho dz' - \mathbf{F}_m - \frac{\sigma}{\rho}(\mathbf{v}_\sigma - \mathbf{v})\right] dz \\ +gH\nabla_{xy}\zeta + \mathbf{f}\times H\bar{\mathbf{v}} - \tau_0 H\bar{\mathbf{v}} - H\mathbf{\Psi} + C_d|\mathbf{v}_b|\mathbf{v}_b\Bigg] = \left(\frac{\partial}{\partial t} + \tau_0\right)\left[\int_{-h}^{\zeta}\frac{\sigma}{\rho}dz + (P-E)\right]\end{aligned} \tag{20}$$

We solve (20) for ζ instead of the primitive equation (6) in order to preserve established gravity-wave performance on simple elements [Lynch and Gray, 1979]. The only departure here from the 3-D wave equation in Lynch and Werner [1991] is the detailed form of the convective terms, which is similar to the 2-D form used by Luettich *et al.* [1992] and Kolar *et al.* [1994]. In 3-D we use the identity

$$\frac{\partial}{\partial t}\int_{-h}^{\zeta}\mathbf{v}dz \equiv \mathbf{v}|_{z=\zeta}\frac{\partial\zeta}{\partial t} + \int_{-h}^{\zeta}\frac{\partial\mathbf{v}}{\partial t}\,dz \tag{21}$$

in conjunction with the momentum equation, en route to (19) and (20).

8.5 Vertical Meshing

The vertical mesh is adjusted in tidal time to track the free surface. Interior node deployment is arbitrary and may be selected on case-specific criteria. The time-dependent mesh deformation is accounted for as in Lynch and Werner [1991]. In all of the runs reported here, we use a smoothly graded mesh obtained with the transformation

$$z(\varepsilon) = -h + \varepsilon(h+\zeta) - \beta\sin(2\pi\varepsilon) \tag{22}$$

wherein ε increases linearly with node number, from 0 at bottom to 1 at the free surface. The constant β is set to give the desired boundary layer resolution. Recall that the bottommost node, $z = z_b = -h$, lies $\xi_b = 1$ m above the true bottom. We find as a rule of thumb that good representation of the bottom boundary layer is obtained with the bottom element approximately the same size as ξ_b [Naimie, 1995].

8.6 Baroclinic Pressure Gradient

The baroclinic pressure gradient is computed utilizing level surfaces as in Naimie *et al.* [1994]. The density is interpolated from the FEM vertical grid onto level triangular grids at fixed depths (10 m spacing from surface to 60 m; 15 m spacing to 75 m; 25 m spacing to 250 m; 50 m spacing to 500 m; and 100 m below). $\nabla_{xy}\rho$ is computed on these surfaces and interpolated back to the original vertical grid, where the appropriate vertical integrations are performed.

8.7 Solution Procedure

The method of solution for ζ and the horizontal velocity is essentially unchanged from Lynch and Werner [1991]. It is based on the Galerkin weak-form of the governing equations, with nodal quadrature for evaluation of the inner products. A semi-implicit time-stepping procedure is used to solve the implicit wave equation for elevation. This involves three time-levels with implicit, centered gravity waves and centered explicit nonlinearities. The essential computational act is the solution of a sparse, symmetric, positive-definite 2-D FEM matrix which is time-invariant. This is followed by a two time-level solution for velocity with rotation, barotropic pressure, vertical shear, bottom stress, and source terms $\frac{\sigma}{\rho}\mathbf{v}$ centered and implicit in time, and all other terms lagged and explicit. The use of nodal quadrature reduces this calculation to a simple tridiagonal matrix inversion. The vertical variation of viscosity (and diffusivities) is represented by element-wise constants which are recomputed at the beginning of each time step and held fixed during the time step.

The discretization and solution of the T, S, q^2, and $q^2 l$ equations is analogous to that for velocity. The decay terms in the q^2 and $q^2 l$ equations are treated in each time step as quasi-linear first-order decays, and handled implicitly, along with vertical diffusion in general, to avoid stability constraints. These variables therefore require only tridiagonal matrix solution in each time step, like the horizontal velocity.

The vertical velocity is obtained as in Lynch and Naimie [1993] from the z-derivative of the continuity equation. With nodal quadrature, this calculation is also tridiagonal. Finally, the mesh is adjusted in every time step to track the movement of the free surface and distribute it among the nodes at depth according to e.g. equation (22).

9 Appendix: Data File Standards Memo

NUMERICAL METHODS LABORATORY
DATA FILE STANDARDS FOR THE GULF OF MAINE PROJECT

Christopher E. Naimie

March 29, 1993[1]

(Revised by Justin T. C. Ip, January 7, 1994)

Units

All standard data files will be written in MKS units, unless explicitly declared otherwise.

When spherical coordinates are used, the right-handed $(\lambda, \phi, z) = (\text{longitude}, \text{latitude}, z)$ coordinate system will be used, with (λ, ϕ) defined in fractional degrees. Longitude is relative to Greenwich, positive east, and is therefore *negative* in North America; latitude is relative to the equator, positive north.

(x, y) coordinates will be derived from (λ, ϕ) via Mercator projection, centered at the Boston tide gage $(\lambda, \phi) = (-71.03^o, 42.35^o)$. The earth's radius for this purpose is $R = 6.3675 \times 10^6$m.

Tidal phase ϕ will be reported as phaselag, in degrees, relative to Greenwich.

Tidal frequencies ω will be reported in radians/sec (i.e. MKS). For example, the M2 frequency based on 12.42 hour period is $\omega = 1.4053 \times 10^{-4}$/sec.

Frequency - Time Domain

In reconstructing time series, the formula is:

$$f(t) = a\,cos(\omega t - \frac{\pi}{180}\phi)$$

with a the real-valued amplitude. Equivalently, in complex exponential form,

$$f(t) = Re(A\,e^{j\omega t})$$

with the complex amplitude $A = a\,e^{-(\frac{j\pi}{180}\phi)}$ and $j = \sqrt{-1}$

File Format/Naming

All files will be machine-readable in ASCII (*) format, unless otherwise indicated.

NML standard data files for the Gulf of Maine project are designated by descriptive three-character trailing sequences that are preceded by a period (.).

---

[1] This is an update of NML memo (DRL,11/11/91)

Cartesian Geometry Files

Several standard files will be used to define only geometry. These will all share a common meshname, with a three-character trailing sequence identifying the type of file. For example, the file "**gom1.nod**" would indicate a node file for the gom1 mesh.

meshname.README: Self-explanatory.

meshname.nod: The **.nod** file contains (x,y) pairs for all nodes in a given mesh. There is no header; the file contains one line of the following form for each node I in the mesh.

I X(I) Y(I)

meshname.ele: The **.ele** file contains triangle incidence lists for all elements in a given mesh. There is no header; the file contains one line of the following form for each element L in the mesh. The incidence list is required to be in counterclockwise order.

L IN(1,L) IN(2,L) IN(3,L)

meshname.bat: The **.bat** file contains bathymetry data for all nodes in a given mesh. Bathymetric depth is positive. There is no header; the file contains one line of the following form for each node I in the mesh.

I H(I)

meshname.lnd: The **.lnd** "land node" file is used for display purposes only. Together with the **.lel** "land element" file, it describes a mesh of triangles covering the land areas adjacent to the relevant hydrodynamic mesh and conforming to it along the land-water interface. The **.lnd** file contains (x,y) pairs for all nodes in a given land mesh. There is no header. The format is identical to the **.nod** format.

meshname.lel: The **.lel** "land element" file contains triangle incidence lists for all elements in a given land mesh. There is no header. The format is identical to the **.ele** format.

meshname.nei: The **.nei** file conforms to TRIGRID specifications. There are three header lines:

line 1: the total number of nodes
line 2: the maximum number of neighbors for any node (MX below)
line 3: the range of the (x,y) data in the file: xmax, ymax, xmin, ymin
Following these lines, there are NN lines of the form:

I X(I) Y(I) NCD(I) H(I) [NGH(I,J), J=1,MX]

where:
X(I),Y(I) are the cartesian coordinates of node I
NCD(I) is the node type code of node I
H(I) is the bathymetric depth at node I
NGH(I,J) are the nodes in the mesh that are "neighbors" to node I
I indexes from 1 to NN, where NN is the number of horizontal nodes

In the FUNDY4 Users' Manual (Lynch, 1990), unique node type codes from 0 to 11 are defined, reflecting various physical boundary conditions. For the purposes of describing geometry, only three codes are relevant:

0: interior
1: exterior boundary
2: island boundary

All publicly-available .nei files in Dartmouth directories will be restricted to these values of NCD(I), preserving their universality as geometry files. (See the description of **.bel** files below.) The program CONVCODES is available for converting between file types.

meshname.gr2: A geometry file with various node and element connectivities pre-computed, for use with the DROG3D simulator (Blanton (1992)). These are derived from the above files by the program CONNECT2D. They contain only geometry data.

Spherical Geometry Files

Several additional standard files contain the equivalent geometry in (λ, ϕ) coordinates. These are indicated by the trailing characters "**ll**". For example, "**gom1ll.nod**" indicates a node file in spherical coordinates for the gom1 mesh.

meshnamell.nod: spherical equivalent of meshname.nod

meshnamell.lnd: spherical equivalent of meshname.lnd

meshnamell.nei: spherical equivalent of meshname.nei

meshnamell.gr2: spherical equivalent of meshname.gr2

Boundary Condition Files

These files indicate physical/dynamic conditions to be enforced on a given mesh. There will be several such files for a given geometry, each representing different combinations of physical forcing at the boundaries.

.bel: A boundary element description of the boundary conditions. There is a two-line header:

line 1: the geometric meshname
line 2: arbitrary text identifying the contents of the file
Following these lines, there are NBE lines of the form:

L IN(1,L) IN(2,L) IBC(left) IBC(right)

where: element L is a line segment beginning at node IN(1,L) and ending at node IN(2,L)
IBC(1) indicates the nature of the boundary on the left of the element
IBC(2) indicates the nature of the boundary on the right of the element
IN(I,L) is a conventional incidence list for 1-D, linear boundary elements
L indexes from 1 to NBE, where NBE is the number of boundary elements

The node numbers refer to the nodes in the triangular mesh file **meshname.nod**, identified in the first header line. The boundary codes are as given in the FUNDY4 Users' Manual [Lynch, 1990]:

0: interior
1: land
2: island
3: nonzero normal velocity
4: geostrophic outflow
5: elevation
6: corner: elevation with land or island
7: corner: geostrophic with land or island
8: corner: nonzero normal velocity with land or island
9: corner: elevation with geostrophic
10: corner: both components of velocity = zero
11: corner: both components of velocity nonzero

Only codes 0-5 are required to describe boundary elements; the additional corner codes are needed only for boundary node classification; they never appear in **.bel** files.

Input Files

.inp: An input file for FUNDY5. This file contains both geometric and boundary condition information, sufficient to obtain a hydrodynamic solution with FUNDY5. It is written by the program TRIGTO-FUNDY4 or CONVCODES, as described in the FUNDY5 Users' Manual.

.din: An input file for DROG3D. This file contains sufficient information to open a velocity output **.vel** file, which in turn contains the geometry meshname, initial positions for one or more drogues, and time-integration parameters for drogue trajectories. It is described in the DROG3D Users' Manual (Blanton (1992)). **Note: time integration parameters in this file are to be given in hours**, an important exception to the general MKS standard.

FUNDY5 Density Data Files

.lst: A 3-D file which contains level surface σ_t density data. There are 4 header lines:

line 1: the geometric meshname
line 2: reserved for user's description of the file
line 3: NLEV, the number of vertical levels for which σ_t data exists.
line 4: The height of each level surface, from the bottom to the top.

$$Z(1) \quad Z(2) \quad Z(3) \quad \ldots\ldots \quad Z(NLEV)$$

Following these lines, there are NN*NLEV lines of the form:

$$\sigma_t(I,L)$$

where: σ_t is the density at horizontal node I, vertical level L (sigmat units)
the inner loop is over 1-D vertical levels; L=1,NLEV
the outer loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the 2-D horizontal mesh

.rho: A 3-D file which contains the baroclinic forcing data required by FUNDY5.

line 1: the geometric meshname
line 2: reserved for user's description of the file
line 3: NNV, the number of vertical sigma mesh nodes under each horizontal 2-D node
Following these lines, there are NN*NNV lines of the form:

I Z(I,J) RHOX(I,J) RHOY(I,J)

where: RHOX(I,J)= $-\frac{g}{\rho_{ref}}\int_{Z(I,J)}^{0}\frac{\partial\rho}{\partial x}dz$

RHOY(I,J)= $-\frac{g}{\rho_{ref}}\int_{Z(I,J)}^{0}\frac{\partial\rho}{\partial y}dz$

the inner loop is over 1-D vertical nodes; J=1,NNV
the outer loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the 2-D horizontal mesh

Following line: elemental values of hrbar: k,hrbarelx,hrbarely
Following NE lines:

K HRBARELX(K) HRBARELY(K)

where: HRBARELX(K)= $-\frac{g}{\rho_{ref}}\int_{Z(K,1)}^{0}[\int_{Z(K,J)}^{0}\frac{\partial\rho}{\partial x}dz]dz$

HRBARELY(K)= $-\frac{g}{\rho_{ref}}\int_{Z(K,1)}^{0}[\int_{Z(K,J)}^{0}\frac{\partial\rho}{\partial y}dz]dz$

the loop is over 2-D horizontal elements; K=1,NE
NE = the number of elements in the 2-D horizontal mesh

Result Files Conforming to DROG3D Standards

.vel: A 3-D complex velocity file describing a velocity output in harmonic form. This file conforms to DROG3D input standards (See Blanton (1992)). **Note that phaselag in .vel files is in radians.** There are 4 header lines:

line 1: the geometric meshname
line 2: reserved for user's description of the file
line 3: NNV, the number of vertical sigma mesh nodes under each horizontal 2-D node
line 4: the number of harmonic constituents contained in the file
Following these lines there are 1 or more blocks containing 3-D velocity amplitude and phase results for a particular harmonic constituent plus its frequency:

Following line: frequency (radians/second)
Following NN*NNV lines:

I Z(I,J) Uamp(I,J) Upha(I,J) Vamp(I,J) Vpha(I,J) Wamp(I,J) Wpha(I,J)

where: Z(I,J) is the vertical coordinate at horizontal node I, vertical node J (MKS)
(Uamp,Vamp,Wamp) are amplitudes of the (x,y,z) velocities (MKS)
(Upha,Vpha,Wpha) are the phaselags of the (x,y,z) velocities (radians)
the inner loop is over 1-D vertical nodes; J=1,NNV
the outer loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the horizontal 2-D mesh

.pth: An output file from DROG3D, describing the trajectories of one or more drogues. There are several header lines:

line 1: the geometric meshname where the relevant **.nod, .ele, .bat, .gr2, etc.** files can be found. For example, "gom1".

lines 2ff: an echo of the **.din** file which controlled the execution of DROG3D and the writing of the present **.pth** file. This echo begins with the **.vel** filename used in the computation of the trajectories.

Following this echo, a separate line with the characters "XXXX" indicates the end of the header section. The balance of the **.pth** file contains the trajectory information as described in Blanton (1992).

Other Standard Result Files

.v2r: A 2-D, real-valued horizontal velocity field, sampled for example at the surface or the bottom of a 3-D mesh. There are 2 header lines:
line 1: the geometric meshname
line 2: reserved for user's description of the file
Following these lines, there are NN lines of the form:

I U(I) V(I)

where: (U,V) are the (x,y) components of horizontal velocity (MKS)
the loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the horizontal 2-D mesh

.v2c: A 2-D, complex-valued horizontal velocity field, sampled for example at the surface or the bottom of a 3-D mesh. There are 3 header lines:
line 1: the geometric meshname
line 2: reserved for user's description of the file
line 3: frequency (radians/second)
Following these lines, there are NN lines of the form:

I Uamp(I) Upha(I) Vamp(I) Vpha(I)

where: (Uamp,Vamp) are the amplitudes of the (x,y) velocities (MKS)
(Upha,Vpha) are the phaselags of the (x,y) velocities (degrees)
the loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the 2-D horizontal mesh

.s2r: A 2-D, real-valued scalar field (e.g. tidal amplitude). There are 2 header lines:

line 1: the geometric meshname
line 2: reserved for user's description of the file
Following these lines, there are NN lines of the form:

I S(I)

where: S(I) is the scalar value at node I (MKS)
the loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the horizontal 2-D mesh

.s2c: A 2-D, complex-valued scalar field (e.g. tidal amplitude and phase). There are 3 header lines:

line 1: the geometric meshname
line 2: reserved for user's description of the file
line 3: frequency (radians/second)
Following these lines, there are NN lines of the form:

I Samp(I) Spha(I)

where: Samp(I) is the amplitude at node I (MKS)
Spha(I) is the phaselag at node I (degrees)
the loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the 2-D horizontal mesh

.s3r: A 3-D, real-valued scalar field (e.g. density). There are 3 header lines:

line 1: the geometric meshname
line 2: reserved for user's description of the file
line 3: NNV, the number of vertical sigma mesh nodes under each horizontal 2-D node
Following these lines, there are NN*NNV lines of the form:

I Z(I,J) S(I,J)

where: Z(I,J) is the vertical co-ordinate at horizontal node I, vertical node J (MKS)
S(I,J) is the scalar value at horizontal node I, vertical node J (MKS)
the inner loop is over 1-D vertical nodes; J=1,NNV
the outer loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the 2-D horizontal mesh

.s3c: A 3-D, complex-valued scalar field (e.g. pressure amplitude and phase). There are 4 header lines:

line 1: the geometric meshname
line 2: reserved for user's description of the file
line 3: NNV, the number of vertical sigma mesh nodes under each horizontal 2-D node
line 4: frequency (radians/second)
Following these lines, there are NN*NNV lines of the form:

I Z(I,J) Samp(I,J) Spha(I,J)

where: Z(I,J) is the vertical coordinate at horizontal node I, vertical node J (MKS)
Samp(I,J) is the amplitude at horizontal node I, vertical node J (MKS)
Spha(I,J) is the phaselag at horizontal node I, vertical node J (degrees)
the inner loop is over 1-D vertical nodes; J=1,NNV
the outer loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the 2-D horizontal mesh

.v3r: A 3-D, real-valued velocity field. There are 3 header lines:

line 1: the geometric meshname
line 2: reserved for user's description of the file
line 3: NNV, the number of vertical sigma mesh nodes under each horizontal 2-D node
Following these lines, there are NN*NNV lines of the form:

I Z(I,J) U(I,J) V(I,J) W(I,J)

where: Z(I,J) is the vertical coordinate at horizontal node I, vertical node J (MKS)
(U,V,W) are the real (x,y,z) vector components of the 3-D velocity
the inner loop is over 1-D vertical nodes; J=1,NNV
the outer loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the 2-D horizontal mesh

.v3c: A 3-D, complex-valued vector field. There are 4 header lines:

line 1: the geometric meshname
line 2: reserved for user's description of the file
line 3: NNV, the number of vertical sigma mesh nodes under each horizontal 2-D node
line 4: frequency (radians/second)
Following these lines, there are NN*NNV lines of the form:

I Z(I,J) Uamp(I,J) Upha(I,J) Vamp(I,J) Vpha(I,J) Wamp(I,J) Wpha(I,J)

where: Z(I,J) is the vertical coordinate at horizontal node I, vertical node J (MKS)
(Uamp,Vamp,Wamp) are (x,y,z) velocity amplitudes (MKS)
(Upha,Vpha,Wpha) are (x,y,z) phaselags (degrees)
the inner loop is over 1-D vertical nodes; J=1,NNV
the outer loop is over 2-D horizontal nodes; I=1,NN
NN = the number of nodes in the 2-D horizontal mesh

Note - The **.v3c** format has phase in degrees while the related **.vel** files are in radians.

Input-Output File Specifications Conforming to QUODDY

Input "`.inq`" file

The input file is a formatted `ASCII` data file conforming to the input requirements of QUODDY1, QUODDY2, and QUODDY3 [*Ip and Lynch*, 1994]. The input file with "`.inq`" contains filenames of all of the mesh data required to run the QUODDY models and all the required input program variables.

In constructing the "`.inq`" input file, every line must start in the first column. The comments following one or more blank characters and beginning with either a "`(`" or a "`[`" character after the data input on each line are ignored except the first line of the file. Each input line of the "`.inq`" file is described as the following:

- Line 1 should read the character string, "`Comment:`", which is the label for the input data given in the next line and is ignored during data input.
- Line 2 inputs a one line comment of maximum 72 characters about the current simulation.
- Line 3 should read the character string, "`Case name:`", which is the label for the input data given in the next line and is ignored during data input.
- Line 4 inputs the case name, `CASE_NAME`, of the simulation of maximum 72 characters long with no blank character, and the input character string must start in the first column. With the `CASE_NAME` specified, it is assumed that there exist three mesh data files (symbolic linkage files can also be set up in UNIX environments) prefixed with `CASE_NAME` in the current directory. These required mesh data files must conform to LABLIB standard (Appendix B of the FUNDY5 User's Manual [*Naimie and Lynch*, 1993]). The total numbers of nodes and elements are determined by QUODDY during execution from reading and counting the data in the `CASE_NAME.nod` and the `CASE_NAME.ele` files respectively.
- Line 5 should read the character string, "`Boundary element incidence list:`", which is the label for the input data given in the next line and is ignored during data input.
- Line 6 inputs the file name, `BEL_FILE`, of the boundary element incidence list of the horizontal mesh. `BEL_FILE` must be of maximum 72 characters long containing no blank character, and the input character string must start in the first column. The data file must conform to LABLIB standard for the "`.bel`" files. The total number of boundary elements is determined by QUODDY during execution from reading and counting the data in the `BEL_FILE` file.
- Line 7 should read the character string, "`Initial condition file:`", which is the label for the input data given in the next line and is ignored during data input.
- Line 8 inputs the character string, `ICQ_FILE`, of maximum 72 characters long with no blank character, and the input character string must start in the first column. There are two possible entries of `ICQ_FILE` which govern the start of the simulation:
  - If the character string "`COLD_START`" is specified, the simulation is started at rest. All the field variables are initialized either to zero or to a prescribed initial value and no initial condition file is needed.
  - If a file name with suffix "`.icq1`", "`.icq2`", or "`.icq3`" is specified, the simulation is a "hot start". On execution, QUODDY1, QUODDY2, or QUODDY3 will read the respective initial condition file, `ICQ_FILE` (a "`.icq1`", "`.icq2`", or "`.icq3`" file), and load the initial field variables according to the format specified by subroutine `ICQ` (to be described in the next section).
- Line 9 should read the character string, "`Echo file:`", which is the label for the input data given in the next line and is ignored during data input.
- Line 10 inputs the echo file name, `ECHO_FILE`, of maximum 72 characters long containing no blank character, and the input character string must start in the first column.
- Line 11 should read the character string, "`Simulation parameters:`", which is the general label for the input data given in the subsequent lines and this label line is ignored during data input.

- Line 12 inputs the dimensional units used in the simulation. For the QUODDY series, only SI units are allowed, therefore, line 6 should read "`SI UNITS`" starting from the first column. Otherwise, execution will halt and an error message will appear in the echo file.
- Line 13 inputs three entries of scaling factors for the x, y, and z coordinates respectively.
- Line 14 is the degrees latitude of the center of the mesh (e.g., 43.500 for the g2s mesh).
- Line 15 inputs the minimum depth. Nodal bathymetric depth will be altered to meet this minimum.
- Line 16 gives the starting time of the simulation in hours.
- Line 17 gives the length of the simulation in hours.
- Line 18 gives the time step increment in hours.
- Line 19 gives the maximum number of time steps allowed for the current simulation.
- Line 20 inputs two integers. The first is the number of time steps between calls to subroutine `OUTPUTQ1`, `OUTPUTQ2`, or `OUTPUTQ3`, and the second is the first time step when the output subroutine will be called.
- Line 21 inputs an integer key, `NONLIN`, of either 0 or 1, for switching off or on the nonlinear convective terms. For `NONLIN = 1`, the nonlinear convective terms are included in the calculation.
- Line 22 inputs the implicit gravity wave factor, `THETA`, between 0 and 1; `THETA = 1` for implicit, `THETA = 0` for explicit. `THETA = 0.75` is used for the sample case.
- Line 23 inputs the weighting factor, `TAU0 =` 2×10^{-4}`sec`, for the continuity equation in the shallow water wave equation.
- Line 24 is either the value of the spatially uniform horizontal viscosity (in MKS) or the Smagorinsky parameter used in the velocity dependent horizontal viscosity (a Smagorinsky parameter of 0.28 is used in [*Smagorinsky*, 1963]).
- Line 25 inputs the shoreline slip coefficient between 0 (free slip) and 1 (no slip).
- Line 26 inputs the total number of vertical nodes under each horizontal node.
- Line 27 inputs the vertical viscosity coefficient, N_{e0}, in the Davies-Furnes turbulence closure scheme, i.e., $N_e = N_{e0}\bar{v}^2$, where $\bar{v}$ is vertically averaged velocity. $N_{e0} = 0.2$ is used in [*Davies and Furnes*, 1980].
- Line 28 inputs the dimensionless quadratic bottom stress drag coefficient, e.g., $C_d = 0.005$ is used in the sample run.
- Line 29 inputs the minimum value of the vertical viscosity at any point in time.
- Line 30 inputs the minimum value of the instantaneous bottom slip coefficient, $K_a = C_d v_{bot}$, where C_d is the dimensionless bottom stress drag coefficient, and v_{bot} is the speed at the bottom.
- Line 31 inputs an integer switch, `NLBS`, for choosing the nonlinear bottom stress: `NLBS = 0` for linear stress; `NLBS = 1` for quadratic stress.
- Line 32 inputs the time-weighting parameter, `EPSN` for vertical viscous stress: `EPSN = 1` for the fully implicit case; `EPSN = 0.5` for Crank-Nicholson.

- Line 33 inputs an integer key, `ISMAG`, of either 0 or 1, to signify the type of horizontal viscosity employed. For `ISMAG = 0`, uniform horizontal viscosity is used; for `ISMAG = 1`, Smagorinsky type velocity dependent horizontal viscosity is implemented.

Output "`.icq1`", "`.icq2`", and "`.icq3`", files for hot-start

The output files with, "`.icq1`", "`.icq2`", and "`.icq3`" for hot-start are formatted `ACSII` data files conforming to the input requirements of the initial conditions for QUODDY1, QUODDY2, and QUODDY3 respectively. These hot-start files contain a snapshot of the current state of the simulation required to restart the simulation at the current time.

The exact format of these "`.icq`" hot-start files is as follows:

- Line 1 lists the model name. For the "`.icq1`" file, only the character string "`QUODDY1`", signifying the initial condition input for the QUODDY1 model, is allowed. Other character strings will halt the simulation. For the "`.icq2`" file, either "`QUODDY1`" or "`QUODDY2`" is allowed to accept either QUODDY1 or QUODDY2 type of initial condition state. For the "`.icq3`" file, "`QUODDY1`", "`QUODDY2`", or "`QUODDY3`" is allowed to accept any one of the three types of initial condition state.
- Line 2 lists the case name, `CASE_NAME`, of the simulation.
- Line 3 gives the input "`.inq`" file of the simulation which created the current hot-start file.
- Line 4 gives the core program which created the current hot-start file.
- Line 5 writes the initial condition state of the simulation which created the current hot-start file.
- Line 6 inputs two integers. The first is the number of nodes in the horizontal mesh, and the second is the number of vertical nodes.
- Line 7 inputs an integer and two real numbers. First, the integer is the number of time steps the simulation has taken to create the current hot-start file. The two real numbers following are the current and the previous simulation time (in seconds) of this initial state.
- The next `NN` lines give the vertically averaged free surface elevation and velocity at the current and the previous time step. Each line is of the form

  ```
  HMID(I)  UMID(I)  VMID(I)  HOLD(I)  UOLD(I)  VOLD(I)
  ```

- The next `NN`×`NNV` lines give the full 3-D description of all the field variables of the current state required for hot-start. The length of each line depends on the hot-start file employing.
  - The remaining "`.icq1`"data can be read by the following `FORTRAN` statement

    ```
    DO J=1,NNV
       DO I=1,NN
          READ (9,*) ZMID(I,J), ZOLD(I,J), UZMID(I,J), VZMID(I,J), WZMID(I,J)
       ENDDO
    ENDDO
    ```

    where `ZMID(I,J)` and `ZOLD(I,J)` are the z-coordinates at the current and the previous simulation times respectively, and `UZMID(I,J)`, `VZMID(I,J)`, and `WZMID(I,J)` are the three components of the 3-D velocity.
  - The remaining "`.icq2`"data can be read by the following `FORTRAN` statement

```
DO J=1,NNV
   DO I=1,NN
      READ (9,*) ZMID(I,J), ZOLD(I,J), UZMID(I,J), VZMID(I,J), WZMID(I,J),
                 Q2MID(I,J), Q2LMID(I,J), RHOMID(I,J)
   ENDDO
ENDDO
```

where `Q2MID(I,J)` and `Q2LMID(I,J)` are the 3-D turbulence variables, q^2 and $q^2 l$ with q and l being the turbulence energy and the mixing length respectively, and `RHOMID(I,J)` is the 3-D density field at the current simulation time.

– The remaining "`.icq3`"data can be read by the following `FORTRAN` statement

```
DO J=1,NNV
   DO I=1,NN
      READ (9,*) ZMID(I,J), ZOLD(I,J), UZMID(I,J), VZMID(I,J), WZMID(I,J),
                 Q2MID(I,J), Q2LMID(I,J), RHOMID(I,J), TMPMID(I,J),
                 SALMID(I,J)
   ENDDO
ENDDO
```

where `TMPMID(I,J)` and `SALMID(I,J)` are the 3-D temperature and salinity fields at the current simulation time respectively.

Output "`.nrv`" file

The output file with, "`.nrv`" is a formatted `ACSII` data file containing a snapshot of a vertical cut of a scalar field variable and velocity. Each "`.nrv`" file contains `NNV`×`NSEG` lines. Each line is of the form

```
I    ABS(I)  ORD(I)  SCAL(I)  VELN(I)  VELT(I)  VELZ(I)
```

where the data vectors, `ABS(I)` and `ORD(I)`, are the abscissa and ordinate of the coordinates of the defined nodes on the plane of the vertical cut, `SCAL(I)` is either the density, or temperature, or salinity, `VELN(I)`, `VELT(I)`, and `VELZ(I)` are the nodal normal, tangential, and vertical components of the velocity on the plane of the vertical cut. The index `I` runs from 1 to `NNV`×`NSEG`, which is the total number of nodes on the plane of the vertical cut, and `NNV` is the number of vertical nodes and `NSEG` is the number of defined horizontal nodes along the vertical cut.

Acknowledgements

We thank our colleagues Christopher Naimie, Francisco Werner, Brian Blanton, Monica Holboke, and Matthias Johnsen for their valuable inputs. Financial support from the National Science Foundation, the Office of Naval Research, the Joint NSF-NOAA GLOBEC program, the Gulf of Maine Regional Marine Research Program, and the New Hampshire Sea Grant College Program is gratefully acknowledged.

References

[*Blanton*, 1987] Blanton, B.O., *DROG3D.f - User's Manual for 3-Dimensional Drogue Tracking on a Finite Element Grid with Linear Finite Elements.* Skidaway Institute of Oceanography, Savannah, Georgia, U.S.A., July 1992.

[*Blumberg and Mellor*, 1987] Blumberg, A F. and G.L. Mellor, "A description of a three-dimensional coastal ocean circulation model", *Three-Dimensional Coastal Models*, N.S. Heaps, ed. American Geophysical Union, Washington D.C. Coastal and Estuarine Series **4**, 1–16, 1987.

[*Blumberg et al.*, 1992] Blumberg, A.F., B. Galperin, and D.J. O'Connor, "Modeling vertical structure of open-channel flows", *ASCE, J. Hydraulic Engg.*, **118**, 1119–1134, 1992.

[*Davies and Furnes*, 1980] Davies, A.M., and G.K. Furnes, "Observed and computed M2 tidal currents in the North Sea", *J. Phys. Oceanogr.*, **10**, 237–257, 1980.

[*Galperin et al.*, 1988] Galperin, B., L.H. Kantha, S. Hassid, and A. Rosati, "A quasi-equilibrium turbulent energy model for geophysical flows", *J. Atmos. Sci.*, **45**, 55–62, 1988.

[*Gill*, 1982] Gill, A.E., *Atmosphere-Ocean Dynamics*, Academic Press, 599–603, 1982.

[*Kinnmark*, 1986] Kinnmark, I.P.E., "The shallow water wave equations: Formulation, analysis and application", *Lecture Notes in Engineering*, **15**, Editors: Brebbia and Orszag, Springer-Verlag, Berlin, 187pp, 1986.

[*Kolar et al.*, 1994] Kolar, R.L., J.J. Westerink, M.E. Cantekin, C.A. Blain, "Aspects of nonlinear simulations using shallow-water models based on the wave continuity equation", *Computers in Fluids*, **23**, 3, 523–538, 1994.

[*Luettich et al.*, 1992] Luettich, R.A. Jr., J.J. Westerink, and N.W. Scheffner, "ADCIRC: An Advanced Three-Dimensional Circulation Model for Shelves Coasts and Estuaries, Report 1: Theory and Methodology of ADCIRC- 2DDI and ADCIRC-3DL", *Dredging Research Program Technical Report DRP-92-6*, U.S. Army Engineers Waterways Experiment Station, Vicksburg, MS, 1992.

[*Lynch*, 1986] Lynch, D.R., "Basic Hydrodynamic Equations for Lakes", *Physics-Based Modeling of Lakes, Reservoirs, and Impoundments*, Chapter 2. W.G. Gray, ed. ASCE, New York 1986.

[*Lynch*, 1990] Lynch, D.R., *Three-Dimensional Diagnostic Model for Baroclinic, Wind-Driven and Tidal Circulation in Shallow Seas; FUNDY4 User's Manual*, Dartmouth College Numerical Methods Laboratory Report, No. NML-90-2, Hanover, New Hampshire, U.S.A., 23pp.

[*Lynch*, 1991] Lynch, D.R., Previous version of this memo dated 11/11/91.

[*Lynch and Gray*, 1979] Lynch, D.R., and W.R. Gray, "A wave equation model for finite element computations", *Computers and Fluids*, **7**, 207–228, 1979.

[*Lynch and Werner*, 1987] Lynch, D.R., and F.E. Werner, "Three-dimensional hydrodynamics on finite elements. Part I: Linearized harmonic model", *Intl. J. Numer. Meths. Fluids*, **7**, 871–909, 1987.

[*Lynch and Werner*, 1991] Lynch, D. R., and F. E. Werner, "Three-dimensional hydrodynamics on finite elements. Part II: Non-linear time-stepping model", *Intl. J. Numer. Meth. Fluids*, **12**, 507–533, 1991.

[*Lynch et al.*, 1991] Lynch, D. R., K. D. Paulsen, and W. Liu, *SPRSPAK, Sparse Matrix Routines for Finite Element Applications*, Numerical Methods Laboratory Report, No. NML-91-4, Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire, 1991.

[*Lynch et al.*, 1992] Lynch, D.R., F.E. Werner, D.A. Greenberg, and J.W. Loder, "Diagnostic Model for Baroclinic and Wind-Driven Circulation in Shallow Seas", *Continental Shelf Res.*, **12**, 37–64, 1992.

[*Lynch and Naimie*, 1993] Lynch, D.R., and C.E. Naimie, "The M2 tide and its residual on the outer banks of the Gulf of Maine", *J. Phys. Oceanogr.*, **23**, 2222-2253, 1993.

[*Lynch et al.*, 1995] Lynch, D.R., J.T.C. Ip, C.E. Naimie, and F.E. Werner, "Comprehensive coastal circulation model with application to the Gulf of Maine", *Continental Shelf Research*, in press 1995.

[*Mellor and Yamada*, 1982] Mellor, G. L., and T. Yamada, "Development of a turbulence closure model for geophysical fluid problems", *Reviews of Geophys. Space Phys.*, **20**, 851–875, 1982.

[*Naimie and Lynch*, 1993] Naimie, C.E., and D.R. Lynch, "Three-Dimensional Diagnostic Model for Baroclinic, Wind-Driven and Tidal Circulation in Shallow Seas – FUNDY5 Users' Manual", *Numerical Methods Laboratory Report NML-93-1*, Dartmouth College, Hanover, New Hampshire, March 1993.

[*Naimie et al.*, 1994] Naimie, C.E., J.W. Loder, and D.R. Lynch, "Seasonal variation in the 3-D residual circulation on Georges Bank", *J. Geophys. Res.*, **99**, C8, 15967-15989, 1994.

[*Naimie*, 1995] Naimie, C.E., "On the modeling of the seasonal variation of the three-dimensional circulation near Georges Bank", *PhD Thesis, Dartmouth College*, Hanover, New Hampshire, 1995.

[*Smagorinsky*, 1963] Smagorinsky, J., "General circulation experiments with the primitive equations I. The basic experiment", *Monthly Weather Review*, **91**, 99–164, 1963. (1993)

[*Wolff*, 1993] Wolff, E.M., *CONVCODES Users' Manual*, Dartmouth College Numerical Methods Laboratory Report, No. NML-93-(in preparation), Hanover, New Hampshire, U.S.A.