User's Manual of

An Operational System of the Individual-Based Particle Tracking Model

by

Zhizhang Yang
James P. Manning
David G. Mountain

Northeast Fisheries Science Center
National Marine Fisheries Service, Woods Hole, MA 02543

Spring 2004

Contents

LISTS OF FIGURES ……..…….iv

Chapter

1 INTRODUCTION…….**1**

2 SYSTEM ORGANIZATION……..……..**2**

2.1 Introduction…….…….2
2.2 Graphic User Interface…….2
2.3 Hydrodynamic Model…….…….2
2.4 Individual-based Particle Tracking Model…….……..3
2.5 Statistical Analysis Module…….4
2.6 Real-time Wind Stress Model……..4

3 MODULE OF GRAPHICAL USER INTERFACE……...……5

3.1 Module Organization……..….5
3.2 Functions of GUI Components…….…….…..5
3.3 Compile and Run…….……8
3.4 Implementation of GUI……..…..9
3.5 Data Files…….……9

.

4 MODULE OF INDIVIDUAL-BASED MODEL……..……11

4.1 Introduction……. ..11
4.2 Source Codes, Input, and Output Files……..11

5 MODULE OF STATISTICAL ANALYSIS……..18

5.1 Introduction……18
5.2 Matlab Scripts……18

6 MODULE OF REAL-TIME WIND STRESS……...…**21**

6.1 Introduction…….…….……..21
6.2 Source Codes and Data Files…….……21
6.3 Program Run…….…….23

7 PROCEDURES OF SYSTEM RUN……..**24**

7.1 Introduction……24
7.2 Procedures of System Run…….24

REFERENCES……..…**27**

LIST OF FIGURES

Figure 2.1 – System Organization ……28

Figure 3.1 – Graphical User Interface for Parameter Specification……...29

Figure 3.2 – Setup Initial Particle Locations……...…….….30

Figure 3.3 – Parameters of Wind Drift Currents…….…..31

Figure 3.4 – Parameters of Velocity Components Due to Random Walks…….…..32

Figure 3.5 – Create File Frame and File Browser Frame…….….33

Figure 3.6 – Graphical User Interface for Displaying Particle Path……..…34

Figure 5.1 – Subregion Definitions……35

Figure 5.2 – Instantaneous Horizontal Locations of Particles……..…….36

Figure 5.3 – Distribution of Particle Horizontal Locations.…….…..….……..37

Figure 5.4 – Vertical Locations of Particle Trajectory……..……..…….….38

Figure 5.5 – Water Temperature along Particle Trajectory……..……..…… …..39

Chapter 1 Introduction

This is a user's manual of an operational system for evaluating the implications of the Lagrangian transport for fish egg/larval in the U.S. northeast coastal region. The system utilizes an individual-based particle-tracking model (IBM) to resolve the transport mechanisms of the biomass based a 3-D flow field and some assumed biological characteristics of the eggs and larvae. The flow fields could be a monthly mean climatological description derived from an unstructured hydrodynamic numerical model, a flow field derived by solely considering the effects of the wind stress, and/or a combination of the two. The whole system is organized in a front-end infrastructure. A graphical user interface (GUI) implemented in Java acts as the front and end, while the actual IBM is conducted by background execution of a set of FORTRAN programs. The system could be used to study the biomass retention and dispersion within certain closed areas and the connectivity among the areas. It is aimed at providing advice for fishery management decisions as well as for the process oriented research.

The manual describes the system’s infrastructure, implementation, and procedures to run the model. The rest of the chapters are organized as follows. Chapter 2 explains model organization and briefs the constituent modules of the system. Chapter 3 describes the implementation of the GUI. Chapter 4 explains the IBM. Chapter 5 describes the module for result analysis. Chapter 6 explains the real-time wind stress module. Chapter 7 describes the procedures to run the package.

Chapter 2 System Organization

2.1 Introduction

The system consists of five modules, namely the graphical user interface, hydrodynamic model, individual-based particle-tracking model, statistical analysis package, and real-time wind stress retrieval package. The former three are sufficient for the model runs in climatological sense. The statistical analysis module is for analyzing the model outputs. The wind stress retrieval module is used to calculate the wind stress using the buoy data; see Figure 2.1 for a flow diagram of the system organization. The modules are briefly described in the following.

2.2 Graphic User Interface

The GUI is implemented in Java. It provides users the interfaces to control the model execution. It allows for specifying the IMB parameters, triggers the execution of the background FORTRAN programs, and displays the results. See chapter 3 for more details.

2.3 Hydrodynamic Model

Knowledge of the 3-D coastal hydrodynamic fields is essential for the particle tracking. Due to the temporal and spatial scarcity of the in-situ measurements, numerical models are now commonly used to resolve the fields. The present system does not directly include any hydrodynamic models, but it requires a prescribed 3-D velocity and temperature fields. The flow fields included with this package are derived from a finite volume coastal ocean model (FVCOM) developed by the Ecosystem Dynamics Modeling Research Group at the School for Marine Science and Technology, University of Massachusetts-Dartmouth (Chen et al. 2003). It is an unstructured grid, prognostic, primitive equations model. It solves momentum, continuity, temperature, salinity, and density equations. The Mellor-Yamada (Mellor and Yamada 1982) level-2.5 turbulence closure scheme is deployed to parameterize the subgrid-scale processes in vertical direction, while in the horizontal direction the mixing processes is parameterized using Smagorinsky's (Smagorinsky 1963) method. As a primitive equation model, the hydrostatic and Boussinnesq approximation are applied.

The model domain covers the U.S. northeast coastal region from the Scotian Shelf in the northeast to Cape Hatteras in the southwest, with an offshore boundary in the slope water; see Figure 3.2 for the model domain definition. It includes the Nova Scotian shelf, Gulf of Maine, Georges Bank, Southern New England, and Mid-Atlantic Bight. A triangular mesh of 10127 nodes and 18784 cells are created to map the domain. The water depths at mesh nodes were interpolated according to the United States Geography Survey (USGS) ETOPO5 data set. The distance between nodes varies with bathymetry between 1.5 km to 47.6 km. It is generally small in the shallow water regions or the areas of sharp bottom slope and large for over open ocean areas. In vertical

direction, a topographic following, sigma-coordinate of 31 evenly distributed layers are used for a smooth representation of irregular bottom. The model run starts from a climatological mass field, i.e., monthly mean temperature (T) and salinity (S), forced by M_2 tide along the open ocean boundary. The T/S climatologies were derived from the T/S profile data between 1912 and 2002 distributed by the Bedford Institute of Oceanography (BIO), Dartmouth, Nova Scotia, Canada. An objective analysis algorithm (OAX) (Henry and He 2002) developed at BIO was applied to finally derive the climatologies. The tidal information was from the FES94-1 global ocean tide Atlas (Lyard and Genco 1994). It includes the harmonic constants on the 0.5° longitude x 0.5° latitude resolution grid and was produced on the base of an finite element hydrodynamic model. Harmonic constants at the model mesh nodes are derived from the tidal atlas via a bilinear interpolation. Flow fields provided with this package are the detided 3-D velocity fields.

2.4 Individual-Based Particle Tracking Model

The IMB explores ecological systems by studying the properties of the individuals constituting them (Werner et al. 2000). Deploying a Lagrangian particle tracking model, the present system is capable of (1) calculating the particle trajectories and the water temperatures along the trajectories based on the 3-D monthly mean velocity and temperature fields from the FV model, or (2) tracking surface drifting objects using the wind drift currents derived from the climatological wind stress or the real-time buoy wind data. In addition, some assumed biological characteristics, e.g., diel migration and preferred residential depths, are used to simulate behaviors of the planktonic fish eggs and larvae. To reflect the random factors due to the diffusions in the flow fields or the biological behaviors, random walks of Gaussian distribution in forms of the horizontal and vertical velocity perturbations are introduced.

The IMB module is organized as follows. A main program in FORTRAN orients the prognostic particle tracking. It loads the wind stress and flow field data, including the 3-D monthly mean velocity ($\mathbf{v}_{model}$), temperature, monthly mean wind stress or the wind stress derived from the real-time buoy data, and the control parameters needed by the IMB; and conducts the actual prognostic particle tracking according to the prescribed flow fields and the particle's biological behaviors. At each time step, the model calculates the wind drift velocities ($\mathbf{v}_{windDrift}$), specifies velocity perturbations ($\mathbf{v}_{random}$) due to the random effects, and obtained the instantaneous velocity field, $\mathbf{v}_{env} = \mathbf{v}_{model} + \mathbf{v}_{windDrift} + \mathbf{v}_{random}$.

The biological characteristics of fish larvae are specified in a 'biological control file' (see section 4.2.2 for the file details), which defines the properties of the larval diel migration. The properties include the migration occurrence time, the vertical swimming rates, and preferred residential depths. The model calculates the vertical motion during each time step of the prognostic model run and superimposes the position change on that solely due to the $\mathbf{v}_{env}$. The diel migration behaviors are specified as follows. By default, larvae move downwards between 4:00 AM and 8:00 AM, and then remain silent till 8:00 PM; after that they move upwards until 4:00 AM of the next morning. The upward and downward swimming rates are set as 1 mm/s and 2 mm/s, respectively. The residential

depth is defined between 3 m depth and the shallower one between the local bathymetry and the 150 m depth. Note that the 150 m depth is about the maximum of the Ekman layer, seasonal thermocline, and euphonic zone. Users can alter the parameter values according to their specific situations. It is noted that the larvae may reach either the water surface or the topographic bottom during the migration. When the former occurs, the larval is set at a depth of 0.01 m below the surface and remains there until a downward velocity is available; when the later happens, it is set at a depth of 0.01 m above the bottom and remains there until an upward velocity is available. A larval is no longer tracked if it is out of the horizontal domain of the model.

2.5 Statistical Analysis Module

The statistical analysis module consists of the Matlab scripts for analyzing the model output. It reads the result file holding time series of the particle locations and the water temperature, retrieves the 4-D locations (longitude, latitude, depth, date/time) of each individual particle and the water temperatures along the particle trajectories, estimates the statistics of the particle horizontal locations, and saves the results as image files.

The statistics are about the geographical distributions of particles among ten predefined regions at some given date/time. By default, the ten regions are the central Georges Bank, Wilkinson Basin, Jordan Basin, Georges Basin, Georges Bank, Mid-Atlantic Bight, Scotian Shelf, Nantucket Shoals, Slope Water Regime, and Gulf of Maine, respectively; see Figure 5.1. Users may alter the region definitions according to their needs. For a given date/time, the percentage of the number of particles in each region is calculated. Images of the instantaneous horizontal locations, their geographical distributions, time series of the vertical location and water temperature along the particle trajectories are output.

2.6 Real-time Wind Stress Module

The wind stress module is to download and calculate wind stress from the real-time buoy wind data. It consists of UNIX and Matlab scripts. The former is used to retrieve the real-time buoy wind velocity data from the web site of the National Buoy and Data Center, while the later is for calculating the wind stress from the buoy data. The neutral stability drag coefficients from Large and Pond (1981) are used for the wind stress calculation.

Chapter 3 Module of Graphical User Interface

3.1 Module Organization

The GUI module is implemented as Java applications. Various Java frames are used for creating the IBM control file, orienting the model run, and displaying results. In the following, the functionalities and implementations of various Java frames are explained. Then, procedures to run the GUI are introduced. Finally, formats of the involved data files are described.

3.2 Functions of GUI Components

3.2.1 Main Frame

The *Main Frame* is first populated when the module is initiated; see Figure 3.1. It is used to create the IMB control file, trigger the model run, and view results. The control file holds the initial temporal and spatial properties of the target particles and parameters to specify the modes of the IBM run. It consists of the following components:

Date/Time The first two rows in *Main Frame* are the text fields for the inputs of the releasing date and time of the target particles. They are listed in the following:

Year: integer value. Since the present system focuses on studying the transportation mechanisms in climatological sense, the value of this field is not critical in the model run.
Month: integer value, 1-12
Day: integer value, l-30 (31)
Hour: integer value, 0-23
Minute: integer value, 0-59
Second: integer value, 0-59
Number of Tracking Days: duration of the model run in days
Time Step (sec): time step of the model run in seconds

Horizontal Locations The particle locations can be specified either manually via the GUI *Map Frame* or via a polygon file; see Figure 3.2. The GUI provides the following three ways for manually specify the particle locations:

(1) To chose the locations individually by clicking various points in *Map Frame;* see Figure 3.2(a);

(2) To specify a region of interest in rectangular shape via the mouse drag within *Map Frame*; see Figure 3.2(b). The spatial separation between the particles is determined by values in the two text fields, *Number of Particles along Lon Dimension* and *Number of Particles along Lat Dimension*, which represent the number of the particles along the zonal and meridianal directions, respectively. In detail, say the (lon0, lat0) and (lon1,

lat1) are the longitudes and latitudes at the top left and lower right corners of the rectangular, respectively; and the N_{lon} and N_{lat} represent the values of the above two text fields. The separations along the zonal and the meridianal directions, (ΔLat, ΔLon), respectively, equal to

$$\Delta Lon = (lon1 - lon0)/(N_{lon} - 1), \text{ and}$$
$$\Delta Lat = (lat1 - lat0)/(N_{lat} - 1).$$

(3) To specify a region of interest in rectangular shape by specifying the boundary longitudes and latitudes at the text fields, *Left Longitude (W)*, *Right Longitude (W)*, *Upper Latitude (N)* ,and *Lower Latitude (N)*, which represent the longitudes and latitudes at the top left and lower right corners of the rectangular, respectively. The remaining is the same as (2).

The other way to specify the region of interest is via an ASCII file which defines a polygon of any arbitrary shape; see Figure 3.2(c). To do this, users need to ignore the above procedures and continues with the followings. The polygon file name is asked after clicking the *Create Ctrl File* button.

Vertical Depth The initial particle depth is specified at the text field, *Particle Depth (m).* The input number can be either positive or negative. Some conversion is done in background. Note that particles of the initial depths below the local bathymetry are treated as being out of the model domain and are not tracked during the IBM run.

Number of Particles at Each Location This is a text field to specify the number of particles ($N_{initial}$) initially at the same location. It is used to evaluate the random effects on the larval migration. This text field takes effects only when $N_{initial} > 1$. Particles initially at the same locations are assigned different velocity fluctuations of the Gaussian distributions at each model step. Hence they may experience different pathways and the random effects on their trajectories can be estimated.

Model Options There are three check boxes for the model options, *Wind Drift Currents, Random Factors,* and *Residual Currents.* Users may select any one of the options individually or any combinations of the three.

Wind Drift Currents Check Box: Check on the box will prompt a frame comprising four check boxes and two text fields; see Figure 3.3 The check boxes are the *Climatology Data, Buoy Data, Object Tracking,* and *Wind Amp Factor,* respectively. The two text fields, namely *Object Tracking* and *Wind Amp Factor,* are two coefficients corresponding to the last two text fields, respectively.

Climatology Data Check Box: use monthly mean climatological wind stress

Buoy Data Check Box: use the real-time buoy wind stress

Object Tracking Check Box: derive the wind drift currents by applying an amplification factor to the wind stress. The factor is the input at the *Object Tracking* Text Field.

Object Tracking Text Field: a constant coefficient to convert the wind stress to

the surface drift currents.

Wind Amp Factor Check Box: calculate the wind drift currents based on the Ekman theory;

Wind Amp Factor Text Field: an arbitrary coefficient as a ratio between the wind stress used in the model run and the real ones. It is for the sensitivity test of the wind stress effects on the particle pathways. The default value is 1.

Random Factors: Checking this box prompts a frame for specifying the maximum magnitude of the horizontal and vertical velocities due to various random factors; see Figure 3.4. The defaults are 5 mm/s for horizontal velocity and 1 mm/s for vertical velocity.

Residual Currents: apply the 3-D residual flow fields during the IBM run.

Done Button: close the frame.

Create Ctrl File A *Create Ctrl File* frame prompts when the *Create Ctrl File* button is clicked. It is for creating the IBM control file. It consists of two check boxes and four buttons; see Figure 3.5. The check boxes are the *Read (lon, lat) Polygon File* and *Use GUI Map for Location*, respectively. They are mutually exclusive. The four buttons are the *Specify Result File Name, Select Bio Ctrl File, Specify IBM Ctrl File Name,* and *Done File Naming.*

Read (lon, lat) Polygon File: Check on this box indicates to specify the region within which particles initially locate according to the vertices coordinates defined in a polygon file. After checking the box, a file browser window prompts for users to select the polygon file (refer section 3.5).

Use GUI Map for Locations: manually specify the initial particle locations as described in the previous paragraphs.

Specify Result File Name: specify the name of the result file. Clicking on this button prompts a file browser window for the filename specification.

Select Bio Ctrl File: select the biological control. Checking this box prompts a file browser window for filename selection.

Done File Naming: close the *Create Ctrl File* frame.

Start Model Tracking Click on *Start Model Tracking* button (Figure 3.1) prompts the *Select IBM Control File* window for users to choose an IBM control File. After this a Unix script file, namely *scriptCall_ibmModel*, are created and executed. See details about the script file in section 3.5. The execution of the script file triggers the FORTRAN program run.

View Result Click on this button prompts the *Select An IMB Result File* window for users to select a result file. The *Result Display Frame* appears right after the selection. The frame provides the functionality to animate the particle horizontal locations.

Quit Model Run Click on this button ends the GUI session and returns Unix command prompt.

(There is a separate standalone document (Yang et al. 2003) available for creating

similar IBM control file. Users are recommended to cross reference that document.)

3.2.2 Frame of Model Domain Map

This frame provides the GUI for users to manually specify the initial horizontal locations of particles. It displays the model domain setup and marks the particle locations after each mouse click; see Figure 3.1. Note that there are four ways to specify the locations, (1) direct mouse click on the map, (2) mouse drag on the map to define a region of interest in the rectangular shape, (3) assigning the longitude and latitude ranges at relevant text fields in *Main Frame*, and (4) data input from a polygon file. The first way specifies the definite locations of each individual particle, while the other three define a region within which the particles initially distribute. Knowing the numbers of particles along the longitudinal and latitudinal dimensions, the actual particle locations are calculated by assuming that the particles are evenly distributed along each dimension.

3.2.3 Frame for Result Display

The *Result display* Frame consists of three components, a canvas panel for animating instantaneous particle locations, a panel of five buttons, and a string displaying date/time information in the format of year/month/date/hour; see Figure 3.6. The button functions are described as follows:

Start: initialize the animation
Pause: freeze the animation at the first click; the second click resumes it
Speed–: slow down animation
Speed+: speed up animation
Quit: quit the display and close the frame

3.3 Compile and Run

To compile the Java source codes, issue the following command in the directory where the Java source codes locate:

> *javac *.java*

> *setenv CLASSPATH nameOfDirectoryHoldThisJavaPackage*[1]

To start the GUI, issue the following command at UNIX/Linux shell prompt:

> *java mFrame*

Please reference chapter 7 for more details about the program run.

3.4 Implementation of GUI

[1] For the machine: ob1.whoi.edu, on which this package is installed, the CLASSPATH is defaulted as /opt/IBM_NMFS.

There are eight Java class files to implement various GUI components in the present system. They are listed in the followings:

DateTimeP.java populates the top portion of *Main Frame* to define the releasing date and time of the target particles. It is labeled as 'Initial Date/Time of Particles' in *Main Frame;* see Figure 3.1.

FileDialogPth.java populates a dialog window for filename specification.

ImgBtnFrm.java populates a frame holding a Java panel to animate the particle horizontal locations along their trajectories; see Figure 3.6.

ImgButtonPanel.java populates a panel for animating the horizontal locations along the particle trajectories; see Figure 3.6. It also populates the command buttons, *Start, Pause, Speed-, Speed+,* and *Quit,* and displays a string showing the date and time information in the format of year/montlh/day/hour. It calls the ImageC.java to populate an image canvas to display particle distributions.

ImageC.java populates a canvas in the *Result Display* Frame for animating distributions of the particle horizontal locations.

LeftPanel.java coordinates the population of the GUI components in *Main Frame,* including the date/time panel, location panel, and the command buttons.

LocationP.java populates the GUI components to specify the initial particle locations and the model mode options. It appears as the middle part of the *Main Frame,* with a title of 'Initial Locations of Particles'; see Figure 3.1.

MapC.java populates the map frame of the model domain; see Figure 3.2. It displays the model domain, mesh setup, and the initial particle locations.

mFrame.java coordinates the population of the *Main Frame;* see Figure 3.1. It calls the classes, LeftPanel.java, DateTimeP.java, LocationP.java, and controls the command buttons, *Create Ctrl File, Start Model Tracking, View Result,* and *Quit Model Run.*

3.5 Data Files

```
(1)Polygon File
    Purpose:
                This file consists of two columns, (longitude, latitude), which
                define the vertices coordinates of a polygon.
    Format:
                (Longitude(k), Latitude(k), k=l,number of vertices)
    Example File:
                obl:/opt/IBM_NMFS/Examples/polygonFile.dat

(2) IBM Control File
                See section 4.2.2
```

(3) Biological Control File
See section 4.2.2

(4) IBM Result File
See section 4.2.3

Chapter 4 Module of Individual Based Model

4.1 Introduction

The individual based model (IBM) conducts the particle tracking. It is implemented in FORTRAN. The model starts from reading the control file (hereafter referred to as the IBM control file) created via the Java GUI. Based on the information in the control file, it loads the wind stress and hydrodynamic flow field data and conducts the prognostic model run. The wind stress could be either the monthly mean climatology from Hellerman and Rosenstein (1983) or that derived from the real-time buoy wind.

The models include the following FORTRAN programs as well as the input and output files in ASCII format:

FORTRAN programs:

Main_ ptr.f
drog4ddt_bio.f
ptr_lib.f
NMLPAKS_000607.f
DCMS.DIM

Input files:

IBM control file
gomMab.bat
gomMab.gr2
fem_windStressClim. dat
fem_z uvwtClim. dat
coefBuoy2Node.dat
buoyList. txt
*yr*Yr*DayTauXY_ xxxxxhyear. dat (xxxxx represents buoy number)*

Output files:

IBM result file
drog.diag

4.2 Source Codes, Input, and Output Files

4.2.1 Source Codes

(1) main_ptr.f

Purpose:

The IBM main program; read the IBM control file, load wind stress and 3-d flow field data, and organize the prognostic IBM run.

Calls:

```
GMT0_GDAY
cstring
read_coefBuoy2Node
```

```
        read_allBuoyWS
        GMT0
        UP_DATE
        GDAY_UTC0
        INTPL_ZUVWT
        INTPL_ATMXY
        Intpl_buoy2node
        CreateRanFem
        DROG4DDT_BIO

Called by:
        scriptCall_ibmMode1 – a UNIX script file created by the Java GUI.
Input:
        fem_windStressClim.dat - montly mean wind stress
        fem_zuvwtC1im.dat - monthly mean surface elevation,3-D velocity, and
        water temperature
        coefBuoy2Node.dat - coefficients to interpolate wind data on the
        model mesh nodes
        buoyList.txt - buoy names and locations
        yrYrDayTauXY_xxxxxhyear.dat – hold time series of wind stress, where
        xxxxx represents the buoy number and year is 2002 at present.
Output:
        Result file whose name is specified in the IBM control file
Note:
        Not all the input files listed in the above are necessarily
        being used in each run. Their being used or not depends on
        the parameter setup in the IBM control file
```

(2) drog4ddt_bio.f

```
Purpose:
        A subroutine conducting particle tracking at each time step. This
        subroutine originated from drog3ddt.f developed by B. Blanton at
        University of North Carolina, NC. It was later renamed to drog4ddt.f
        after revisions done by J. Manning at Woods Hole Laboratory,
        Northeast Fisheries Service Center/NOAA. The drog4ddt.f allows for
        outputting particle locations in longitude and latitude coordinate,
        instead of the coordinate, (x, y), in meters relative to a reference
        location at (69°W,40oN). The drog4ddt_bio.f is developed based on the
        drog4ddt.f. Changes were made to distinguish the particle behavior at
        boundaries of different types, i.e., free surface, bottom, or lateral
        boundaries. Also a subroutine, bioModel_ptr, is incorporated to
        specify the biological characteristics to the originally fully
        passive particles in the previous versions.

Calls:
        BioModel_ptr() in ptr_lib.f
        All the others were originally defined, see Blanton (1995) for
        details.

Called by:
        Main_ptr.f

Input:
        gomMab.gr2 - domain information file. This file is created using
        connect2d.f which was developed by C. Naimie/B. Blanton. For document
        completeness, the connect2d.f is included in the module, though it is
        not directly called by the IBM.
        IBM control file - filename is user-defined
        gomMab.bat - bathymetry

Output:
        IBM result file - see section 4.2.3 for the file structure; Filename
        is specified in the IBM control file.
        drog.diag - diagnostic report file
```

(3) NMLPAKS_000607.f

```
Purpose:
        A subroutine package developed at Numerical Methods Laboratory,
```

```
        Dartmouth College, NH. It provides auxiliary functionalities for the
        routines of the Dartmouth Finite Element Model and present particle
        tracking model. The following routines are used at present. No change
        is made if it is not being addressed explicitly.
Calls:
        N/A
Called by:
        Main_ptr.f, drog4ddt_bio.f
Input:
        See comments in each individual subroutine
Output:
        See comments in each individual subroutine
```

(4) ptr_lib.f

```
Purpose:
        A subroutine/function package developed to include the following
        subroutines/functions.
        bioModel_ptr - the module to implement biological model
        ymd2gday - date conversion from (year, month, day) to Gregorian day
        yDayDec2Int - calculate (hour, minute, second) number in floating
        point format
        yDayInt2Dec - convert (year day, hour, minute, second) in integer
        format to year day as a floating-point number
        intpl_atmxy - temporal interpolation of wind stress data
        ranl - returns a uniform random deviate between O.O and l.0
        gasdev - gasdev returns a normally distributed deviate with zero mean
        and unit variance, using ranl(idum) as the source for the uniform
        deviates
        createRanfem - generate a data set of normally distributed deviate
        with zero mean and variance of 0.5.
        Intpl_zuvw - calculate the surface elevation and 3-D velocity,
        (z,u,v,w), at a given year day from linearly interpolating the
        monthly mean climatology data. This subroutine is obsolete.
        Intpl_zuvwt - calculate the surface elevation, 3-D velocity, and
        water temperature,(z,u,v,w,t), at a given year day from linearly
        interpolating the monthly mean climatological data
        Read_allbuoyWS - read archived buoy wind stress data from multiple
        files; It calls read_buoyws.
        Read_coefbuoy2node - read coefficients for interpolating the wind
        stress data from the buoy locations to the model mesh nodes
        Read_buoyws - read one file holding wind stress data
        intpl_buoy2node - create v3rFn, not used
Calls
        Find_level
        depth_interp
        get_depth
        gday
        gday_utc0
        utc0_gday
        cstring
Called by:
        Main_ptr.f
        drog4ddt_bio.f
Input:
        See comments in each individual subroutine/function
Output:
        See comments in each individual subroutine/function
```

(5) DCMS.DIM

```
Purpose:
        A FORTRAN include file to specify the mesh parameters used in
        drog4ddt_bio.f, and ptr_lib.f
Calls:
        N/A
Called by:
        Main_ptr.f, drog4ddt_bio.f, and ptr_lib.f
Input:
        N/A
```

```
Output:
       N/A
```

(6) makefile

```
make file of main_ptr.f
```

4.2.2 Input Files

(1) IBM Control File[2]

```
Purpose:
         The IBM Control file is created via the Java GUI. It holds
         information of the particle releasing date/time, locations,
         tracking duration, time step, and the model mode options.
Format:
         The following is an example of the IBM control file. Note that the
         comments in [] do not appear in the actual file.

         [1] 7200.0 [model time-step in seconds]
         [2] 5.0    [duration of model run, unit: days]
         [3] 0 1 1 0.0 0.0 0.05 0.0010 5 [see notes]
         [4] userDirectory1/ibmResultFile.dat [model result file]
         [5] userDirectory2/bio_ctrlFile.dat [IBM control file]
         [6] 1 1 [numbers of model time-steps for updating and outputting]
                [the particle locations and output]
         [7] {# grid scaling factors} [comment line]
         [8] 1.0 1.0 1.0  [grid scaling factors in x, y, and z directions]
         [9] {# drog scaling factors} [comment line]
         [10] 1.0 1.0 1.0 [scaling factors of particle coordinates in x, y,
              and z directions]
         [11] {number of starting drogs} [comment line]
         [12] 10                         [as commented on line 11]
         [13] {Starting positions/date (x,y,z, year, yearDay)}
         [14] 97903.47976701148 188497.740643661 -25.0 2002 0.0 -67.85
              41.675 [as commented on line 13]
         [15] ************** END .IND FILE ECHO ***************

         End of File

Note:
         The name of the IBM control file is user-specified (refer section
         7.2)

         Eight numbers on line 3 correspond to
         1)  wind stress flag:
               0: no wind;
               1: climatological wind;
               2: buoy wind
         2) random factor flag:
               0: not considering random effects;
               1: considering random effects
         3)  mass field flag:
               0: not considering residual flows due to the baroclinity
               1: considering residual flows
         4) coefficient to convert the wind stress to water surface
            drift currents
         5) amplification factor of the wind stress
         6) amplitudes of horizontal random velocity fluctuation, unit: m/s
         7) amplitudes of vertical random velocity fluctuation, unit: m/s
         8) number of particles at each location
```

(2)gomMab.bat

```
Purpose:
       Bathymetry at each node
```

[2] Quoddy users might recognize this as a .ind file.

Format:

Node number, bathymetry in meter

(3)gomMab.gr2

Purpose:

Domain information file;
It holds the information of the number of nodes, elements in mesh domain, bathymetry at each node location, list of nodes forming each element, and for each node, a list of its neighboring node

Format:

Number of elements (NE), number of nodes (NN)
(i, x(i), y(i) i=l,NN), where x(i)and y(i)are the x-, y-coordinates of node i)
(j, node(j,l), node(j,2), node(j,3), j=l,NE), where node(j,k) (k=1,2,3) are the node numbers defining the element j
((i, node(i,l), node(i,2),..., node(i,l), ..., node(i,Li)
i=l,NN), l=l,Li), where Li is the maximum number of neighboring nodes of the node i; node(i,l) is the l-th neighboring nodes of node i]

(4)fem_windStressClim.dat

Purpose:

Monthly mean wind stress climatology. For each month, there are 10127 lines corresponding to the nodes in mesh domain. The year days correspond to the middles of each month. The data of the 12 months are concatenated in one file. Besides, for ease of coding the data interpolation in determining the wind stress at a given time step, data of December and January are added at the beginning and the end of the files, respectively and the year days of -15 and 380 are assigned. Therefore, altogether there are 14- month data and 10127x14 = 141778 lines in the file.

Format:

Year, year day, Tx, Ty [As for climatology, Year is treated as a dummy variable, default as ‘0’.]

(5)fem_zuvwtClim.dat

Purpose:

Monthly mean hydrodynamic data, (ζ,u,v,w,t)derived from FVCOM, where ζ, u, v, w, and t are the surface elevation, the three velocity components, and water temperature, respectively. For each month, there are 10127x31 lines of data, where the 10127 and 31 are the number of nodes and the number of vertical layers in mesh of the model domain. The year days correspond to the middles of each month. The data of the 12 months are concatenated in one file. Besides, for ease of coding the data interpolation in determining the (ζ,u,v,w,t) at any given time step, data of December and January are added at the beginning and the end of the file, respectively and the year days of -15 and 380 are assigned. Therefore, altogether there are 14-month data and 2977338 lines in the file.

Format:

Year, year day, ζ, u, v, w, t [As for climatology, Year is treated as a dummy variable, default as '2002'.]

(6)coefBuoy2Node.dat

(see section 6.2.2)

(7)yrYrDayTauXY_xxxxxhyear.dat

(see section 6.2.2)

(8)BIO Control File

Purpose:

Specify the biological characteristics of particles. Its name is user-specified (refer section 7.2)

Format:

The following is an example of the control file. Note that the comments in [] do not appear in the actual file.

```
[ --------------,Beginning of File    ]
[1] Line 1: FileName: bio_CtrlFile.dat [ comment line]
[2] Line 2: Used by: subroutine: bioModel_ptr [comment line]
[3] 9 !number of data lines in this file
[4] 1 !bio. behavior flag; 1: consider diel migration, 0: not.
[5] 18.0 !Hour_up (hour of day) [day hour to start swimming upward]
[6] 6.0 !Hour_down (hour of day) [day hour to start swimming]
                                  [downward]
[7] 9.0 !Hour_stop (hour of day) [day hour to cease active motion]
[8] –1.0 !Z_zoneTop (m) [depth of the top of living layer]
[9] –150.0 ! Z_zoneBot (m) [depth at the bottom of living layer]
[10] 0.0005 !w_upRate (m/s) [upward swimming rate]
[11] -0.0005 !w_downRate (m/s) [downward swimming rate]
[ -------------------------End of File      ]
```

4.2.3 Output Files

(1)IBM result file[1]

Purpose:

Output file from the IBM run. It consists of two parts, (1) an echo of the IBM control file and (2) time series of the particle locations and the water temperatures along the particle trajectories. Its name is user-specified (refer 7.2).

Format:

The following is an example of the result file. Note that the comments in [] do not appear in the actual file.

```
[1] DATA_FILES/gomMab
[2] RECORDS ABOVE THIS LINE IS THE DOMAIN NAME ON WHICH
[3] VELOCITIES WERE COMPUTED
[4]  ************** BEGIN .IND FILE ECHO **************
[5] 3600.0
[6] 50.0
[7] 1 1 1 0.1 0.0050 0.0010 1 2
[8] /workl/result.dat
[9] /workl/bio_ctrlFile.dat
[10] 1 1
[ll] {# grid scaling factors}
[12] 1.0 1.0 1.0
[13] {# drog scaling factors}
[14] 1.0 1.0 1.0
[15] {number of starting positions}
[16] 1
[17] {Starting positions/date (x,y,z, year, yearDay)}
[18] 97903.47976701148 188497.740643661 -25.0 2002 O.O -67.85 41.675
[19] ************** END .IND FILE ECHO ***************
[20] MODEL TIME-STEP (SECONDS)              :      3600.0000
[21] DROG3DDT TIME-STEP (SECONDS)           :      3600.0000
[22] DROG3DDT OUTPUT INTERVAL (SECONDS)     :      3600.0000
[23] XXXX
[24] 0000          3600.000000       1 2
[25] -67.8500 41.6750 -25.0000 -0.3036E+02 7.0824 1 1 2002 0.0833
[26] …
```

Notes:

l)Lines 1-3: comment lines

2)Lines 4-19: echo of the IBM control file; see section 4.2.2 for details of the file structure. Note that there may be multiple lines between lines 18 and 19, depending on the number of particles being tracked.

[1] Quoddy users might recognize this as a .pth file.

3)Lines 20: time step of the IBM run
4)Line 21: number of second to track the particle locations, etc.
5)Line 22: number of second to output the particle locations, etc.
6)line 23: comments;
7)line 24: string '0000', time steps in seconds between sebsequent records, Δtimestep, number of particles initially at different locations within the model domain, N_{difLoc}, number of particles initially at each locations, $N_{eachLoc}$.
8)line 25: Starting from this line, a 9-column matrix is output. The nine columns represent, respectively, the longitude (lon), latitude(lat), drogue depth (z), bathymetry at (lon, lat), water temperature at (lon, lat, z), day, month, year, time lapse in days relative to the time when the model run starts. The length of the matrix (N_{mat}) equals $N_{timestep}\times N_{difLoc}\times N_{eachLoc}$, where $N_{timeStep}$ is the number of total time steps of the model run. It equals to floor($24*3600\times T_{IBM}/\Delta t_{timeStep}$), where T_{IBM} is the model run duration in days and $\Delta t_{timeStep}$ is the model time-step in seconds. The data in the matrix are organized as an output from three-level loops. The outermost one loops through the particles initially at the same locations, ranging from 1 to $N_{eachLoc}$; the intermediate one loops through various output time steps from 1 to $N_{timeStep}$; and the inner loop is about the particles initially at different locations, ranging between 1 and N_{difLoc}.

(2) drog.diag

Purpose:
Hold tracking parameters, run-time diagnostic information of the IBM run.

Format:
See detailed comments in the file

Chapter 5 Module of Statistical Analysis

5.1 Introduction

The statistical analysis module consists of the Matlab scripts for analyzing the IMB output. It reads the IBM result file, retrieves the 4-D locations (longitude, latitude, depth, date/time) of each individual particle and the water temperatures along the particle trajectories, estimates the statistics of the particle's horizontal distributions, and saves the results in image files.

Besides, ten ASCII files are included to define the regions of interest within the model domain. The regions are the central Georges Bank, Wilkinson Basin, Jordan Basin, Georges Basin, Georges Bank, Middle Atlantic Bight, Scotian Shelf, Nantucket Shoals, Slope Water Regime, and Gulf of Main, respectively; see Figure 5.1.

5.2 Matlab Scripts

(l) read_path.m

```
Purpose:
        Read the IBM result file and parse the information therein
Calls:
        N/A
Called by:
        plotDrogZ.m, plotDrogT.m, drogRegionDistr.m
Input:
        Full path/filename of the IBM result file
output:
        gridname - model domain name
        ndrog - number of drogues at start of run
        nts - number of time steps in run
        nRandomRun - number of initial drogues at each locations; it equals
        the number of random runs
        timeStep - IBM time step in seconds
        newPathArr - 4-D array holding result data.
            dimension 1: number of random runs, i.e., number of initial
                         particles at each location
            dimension 2: number of particles of different initial locations
            dimension 3: number of time steps when the model results are
                         output
            dimension 4: 9-column data sets in the format of [lon, lat,
                         depth, bathymetry, temperature, day, month, year,
                         time lapse in days since the particles are released]
```

(2) plotRegionLine.m

```
Purpose:
        Plot regional map of U.S. northeast coastal regions with the ten
        regions outlined
Calls:
        N/A
Called by:
        plotDrog.m
Input:
        The following files are at obl:/opt/IBM_NMFS/Stat/Data.
        necscoast_worldvec.dat
                p_gBankCtr_p.dat
        p_wil_p.dat
        p_jor_p.dat
        p_gBasin_p.dat
```

```
        p_gbank_p.dat
        p_mab_p.dat
        p_sco_p.dat
        p_nan_p.dat
        p_slope_p.dat
        p_gom_p.dat
Output:
        On-screen display of the regional map of U.S. northeast
        coastal regions with the ten regions outlined; see Figure
        5.1.
```

(3) plotDrog.m

```
Purpose:
        Plot snapshoot of the particle regional distribution among
        the ten predefined regions and histograms of the distribution
Calls:
        read_path.m and plotRegionLine.m
Called by:
        N/A
Input:
        Name of the path file, day of interest
Output:
        Image file (.png ) displaying instantaneous horizontal
        locations of particles and the histogram of the distribution;
        see Figures 5.2-3.
```

(4) plotDrogZ.m

```
Purpose:
        Plot time series of vertical positions along the trajectory of one
        individual particle
Calls:
        read_path.m
Called by:
        N/A
Input:
        pathFile - full path/filename of the IBM result file runNumber -
        sequence number of various random runs drogNumber - integer number to
        specify the drogue of interest
        yrDayO - beginning year day on plot
        yrDayl - ending year day on plot
Output:
        Image file (.png) displaying the time series of vertical position
        along the particle trajectories; see Figure 5.4 for an example.
```

(5) plotDrogT.m

```
Purpose:
        Plot time series of the water temperature along the particle
        trajectories
Calls:
        read_path.m
Called by:
        N/A
Input:
        pathFile - full path/filename of the IBM result file
        runNumber - sequence number of various random runs
        drogNumber - integer number to specify the drogue of interest
        yrDay0 - beginning year day to plot
        yrDayl - ending year day to plot
Output:
        Image file displaying the water temperature along the drogue
        trajectories; see Figure 5.5 for an example.
```

(6) drogRegionDistr

```
Purpose:
       From the number of given set of the drogue coordinates, count
       the drogues within each of the ten predefined regions
Calls:
       read_path.m
Called by:
       N/A
Input:
       drogue coordinates in (longitude, latitude)
       lonVec - 1-D array longitude
       latVec - 1-D array, latitude
Output:
       10-element array, each corresponds to the number of drogues
       in one of the following ten regions,
       1: Central Georges Bank
       2: Wilkinson Basin
       3: Jordan Basin
       4: Georges Basin
       5: Georges Bank
       6: Middle Atlantic Bight
       7: Scotian Shelf
       8: Nantucket Shoal s
       9: Slope Water Regime
       10: Gulf of Maine
```

(7) date2yd.m

```
Purpose:
       Convert Gregorian day to year day
Calls:
       N/A
Called by:
       N/A
Input:
       Year, month, day, hour, minute, and second
Output:
       Year, year day, and decimal year day
```

Chapter 6 Module of Real-Time Wind Stress

6.1 Introduction

The real-time wind stress module is for downloading the time series of buoy wind data from the NDBC ftp web site (ftp://seaboard.ndbc.noaa.gov) and calculating the wind stress from the wind velocity. The buoy data include the wind speed and directions at various anemometer heights. Using the logarithmic wind profile relation under the neutral conditions, the wind speeds are first adjusted to those at the l0-meter height above the sea level. The horizontal wind stress (τ) is converted from the wind speed vector data using the neutral wind stress drag coefficients (Large and Pond 1981). The output is a file holding the wind stress time series at the buoy location. It is later used as input to the IBM. Note that the wind stress at the model mesh nodes is actually needed by the IBM. An interpolation from the buoy locations to the mesh nodes is done in the particle tracking module, which utilizes the interpolation coefficients derived from a Matlab script in this package.

It is noted that the quality of the downloaded wind data needs to be seriously controlled. Mainly due to the failure of the buoy anemometer, the downloaded time series are often not continuous records. In some extreme cases, the gap between the two subsequent valid records is large as several months. Therefore, cautious examination of the wind data is suggested before applying them in the model. At present, the buoy data of year 2002 are downloaded and examined. Only the data sets from the anemometers with rare failure are used.

6.2 Source Codes and Data Files

6.2.1 Source Codes

(1) main_ftp.sh

```
Purpose:
        a tc-shell script calling the following two script files
Calls:
        Script_ftp_buoy.sh, cleanData.sh
Input:
        N/A
Output:
        multiple buoy wind data of filenames, xxxxxh2002.txt.gz, where xxxxx
        represents buoy names
```

(2) script_ftp_buoy.sh

```
Purpose:
        Download buoy wind data
Calls:
        N/A
Called by:
        Main_ftp.sh
Output:
        multiple buoy wind data of filenames, xxxxxh2002.txt.gz, where xxxxx
        represents buoy name. The files includes invalid records.
```

(3) cleanData.sh

```
Purpose:
        eliminate the invalid records from the downloaded wind data file,
        xxxxxh2002.txt, where xxxxx represents the buoy name.
Calls:
        N/A
Called by:
        main ftp.sh
Input:
        N/A
Output:
        YmdDirSpd_xxxxx.txt, where xxxxx represents the buoy name
```

(4) buoyWind2Tau.m

```
Purpose:
        convert wind speed/direction to wind stress
Calls:
        stresstc.m from air-sea package of WHOI
Called by:
        N/A
Input:
        buoyList.txt and ymdDirSpd_xxxxx.txt, where xxxxx represents the buoy
        name
Output:
        yrYrDayTauXY_xxxxx.dat, where xxxxx represents the buoy name
```

(5) findDeltaLonLat.m

```
Purpose:
        Create an ASCII file holding coefficients for interpolating wind
        stress from the buoy location to the model mesh nodes
Calls:
        xy211.m, drawelems.m
Called by:
        N/A
Input:
        buoyNumberLatLon.txt
Output:
        coefBuoy2Node.dat
```

(6) AirSea_whoi Package

This is a Matlab script package developed by Bob Beardlsley and Rick Pawlowicz at Woods Hole Oceanographic Institute, MA. It was downloaded from the website, httn://sea-mat.whoi.edu. The package is used to calculate the wind stress from the wind velocity data.

6.2.2 Data Files

(1) xxxxxh2002.txt.gz

gzip file directly downloaded from NDBC ftp web site (ftp://seaboard.ndbc.noaa.gov); see document therein for the data structure

(2) ymdDirSpd_xxxxx.txt

ASCII file holding valid buoy records in the format:
year, month, day, wind direction, wind speed

(3) yrYrDayTauXY_xxxxx.dat

ASCII file holding time series of the wind stress in the format:
year, month, day, τ_x, τ_y, where τ_x and τ_y are the wind stress components in the east and north directions, respectively.

(4) buoyList.txt

prescribed file holding buoy name, location information in the format:
buoy sequence number, buoy file names, latitude, longitude, anemometer height

(5) buoyNumberLatLon.txt

```
prescribed file holding buoy location information in the format:
buoy number, latitude, longitude, anemometer height (m)
```

(6) coefBuoy2Node.dat

```
prescribed file holding coefficients for interpolating wind stress data from
the buoy locations onto the mesh nodes. The format is as follows:
mesh node number, number of buoy data for interpolation (n), list of buoy
sequence number (n values) coefficients (n values)
```

6.3 Program Run

To download the buoy data, issue the following command at the UNIX shell prompt,

Main_ftp.sh

To calculate the wind stress data at buoy locations, issue the following commands at Matlab command prompt,

buoyWind2Tau

To create the auxiliary ASCII file for data interpolation, issue the following commands at the Matlab command prompt,

FindDeltaLonLat

It is noted that users don't need to run the findDeltaLonLat.m. Here it is listed and explained solely for the completeness of the documentation.

Chapter 7 Procedures of System Run

7.1 Introduction

This chapter describes the procedures to run the IBM. First, users need to create an IBM control file (see Chapter 4) using the GUI. Creation of the IBM control file includes specifying of the initial date/time, duration and time steps of IBM run, particle locations at release, model mode options, and biological control file, as well as specifying the names of the IBM control file and result file. Each model run is based on the parameters specified in the particular control file. Users may have many IBM files prepared and can select the suitable one via the GUI. Besides the IBM control file, users may also need other prescribed files, biological control file, polygon file, and wind stress file, depending on the parameter setup in the IBM control file. The former is required if certain biological behaviors are proposed for the target particles. The later is needed if the region of interest is specified via a polygon file (see Chapter 4). Users can alternatively specify the initial particle locations via the Java GUI. After finishing the parameter specification and control file selection, users can start the background FORTRAN programs to actually conduct the particle tracking and save the results in the file whose name is defined in the IBM control file.

7.2 Procedures of System Run

The system has been installed at ob1.wh.whoi.edu: /opt/IBM_NMFS. The procedure is as follows:

Start GUI: Log on the machine, ob1.wh.whoi.edu and issue the following commands from a directory users have written privilege:

tcsh
java mFrame

It initiates the Java GUI; see Figure 3.1.

Create Control File: Go through the following steps to create the IBM control file via the *Main Frame:*

(1) Specify the date/time information at the following text fields:

Year, Month, Day, Hour, Minute, Second,
Number of Tracking Days, and
Time Step(sec)

(2) Specify particle locations:

Horizontal Locations: If users decide to manually specify the locations from

Map Frame, follow the procedures described in section 3.2.1 and define the values of the following text fields:

Particle Depth (m),
Number of Particles along Lon Dimension,
Numberber of Particles along Lat Dimension, and
Number of Particles at Each Location.

If users tend to specify the region of interest from a polygon file, skip this step and wait until step (4).

(3) Select the model options from the following check boxes:

Wind Drift Currents,
Random Factors, and
Residual Currents

Note that a frame for parameter specification appears after clicking either of the *Wind Drift Current* and *Random Factors* check boxes. Please reference section 3.2.1 for details.

(4) Click the *Create Control File* button. There creates the *Create Ctrl File* frame; see Figure 3.1.

Check the box, *Read (lon, lat) Polygon File,* if you wish to specify the region of interest via a polygon file; or else check the box, *GUI Map for Locations,* which uses the information from the GUI manipulation at (2) for particle locations.

Click the button, *Specify Result File Name,* to specify the result filename.

Click the button, *Select BIO Ctrl File,* to provide the name of the biological control file.

Click the button, *Select IBM Ctrl File Name,* to provide the name of the IBM control file. Note that all the above parameter values will be saved in this file.

Click the button, *Done File Naming,* to complete the creation of the IBM control file.

After the above manipulation, the particle locations display on the *Map Frame.* If they are not as expected, redo the 'Create Control File'.

Start Model Tracking: Once the IBM control files are prepared, you are ready to trigger the background FORTRAN programs to conduct the IBM run.

Click the button, *Start Model Tracking,* at the bottom of the *Main Frame.* A frame, *Select IMB Control File,* prompts for selecting an IBM control file. After the

selection, the FORTRAN programs start.

View Result: Click the button, *View Result* and select a result file via the file browser window shown right after the click. The *Frame for Result Display* (see Figure 3.6) prompts. You can manipulate the animation via the buttons on the frame. Depending the length of run, users may expect some delay before viewing the results.

Quit Model Run: Click the button, *Quit Model Run* to quit

References

Blanton, B. O., 1995: DROG3D - User's Manual for 3-Dimensional Drogue Tracking on a Finite Element Grid with Linear Finite Elements. University of North Carolina, Chapel Hill, NC, 13pp.

Chen, C., H. Liu, and R. Beardsley, 2002: An unstructured grid, finite-volume, three-dimensional, primitive equations ocean model: application to coastal ocean and estuaries. *J. Atmos. Oceanic Tech.,* **20,** 159-186.

Naime, C. and D. Lynch, 2003: Drifter Simulation in the Gulf of Maine. Available online at the Website *http://www-nml.dartCmouth.edu/DROG3D/DROG3D.shtml*. Numerical Methods Laboratory, Dartmouth College, Hanover, NH.

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

Hannah, C. G., J. A. Shore and J. W. Loder, 2001: WebDrogue Drift Prediction Model. Available online at the Website *http://www.mar.dfopo.gc.ca/science/ocean/coastal_hydrodynamics/WebDrogue/webdrogue.html*.

Hellerman, S. and M. Rosenstein, 1983: Normal Monthly Wind Stress Over the World Ocean with Error Estimates. *J. Physi. Oceano.,* **13,** 1093-1104.

Hendry, R. and I. He, 2002: Technical Report on Objective Analysis (OA) Project. In. preparation for *Can. Tech. Rep. Hydrog. Ocean Sci.* Available online at the Website *http://www.mar.dfo-mpo.gc.ca/science/ocean/coastal_hydrodynamics/Oax/oax.html*.

Large, W. G., and S. Pond, 1981: Open ocean momentum flux measurements from moderate to strong winds. *J. Phys. Oceanogr.,* **11,** 324-336.

Mellor, G.L., and T. Yamada, 1982: Delopment of a turbulence closure model for geophysical fluid problem. *Rev. Geophys. Space Phys.,* **20,** 851-875.

Werner, F. E., J. A. Quinlan, R. G. Lough, and D. R. Lynch, 2001: Spatially-explicit individual based modeling of marine populations: a review of the advances in the 1990s. *Sasia* **86**: 411-421.

Yang, Z., J. P. Manning, and D. G. Mountain 2003: User's Manual of Java Package for Creating IBM Control File. Northeast Fisheries Science Center, National Marine Fisheries Service, National Oceanic and Atmospheric Administration, Woods Hole, MA, 11pp.

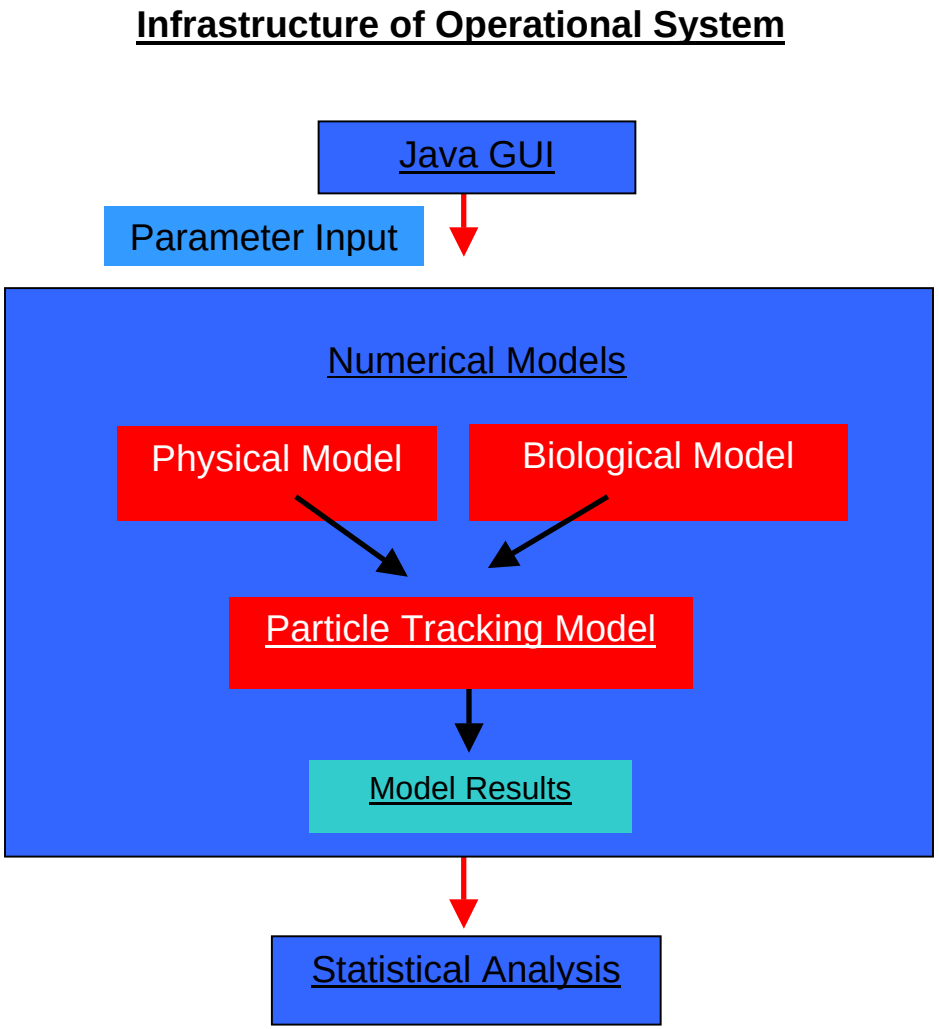

Figure 2.1 - System Organization

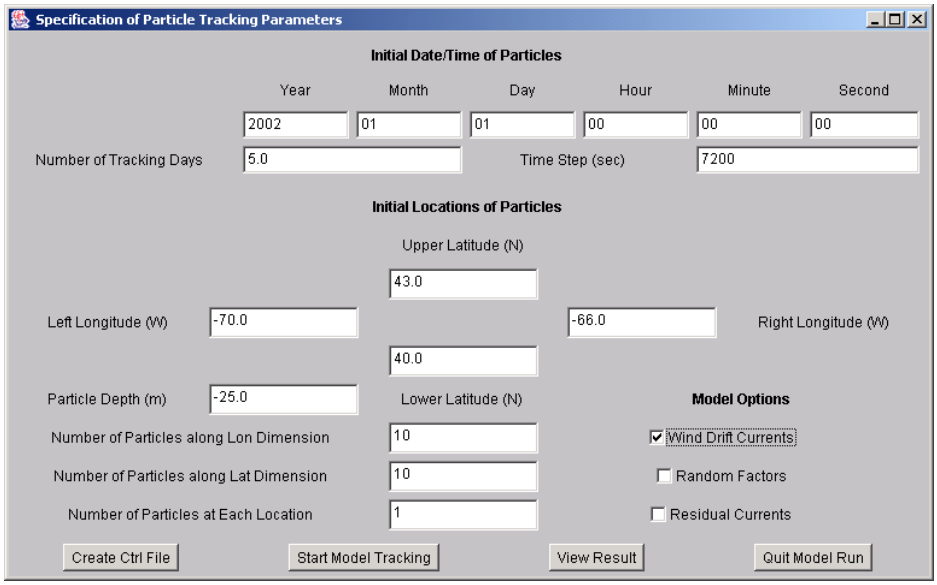

Figure 3.1 – Graphical User Interface for Parameter Specification

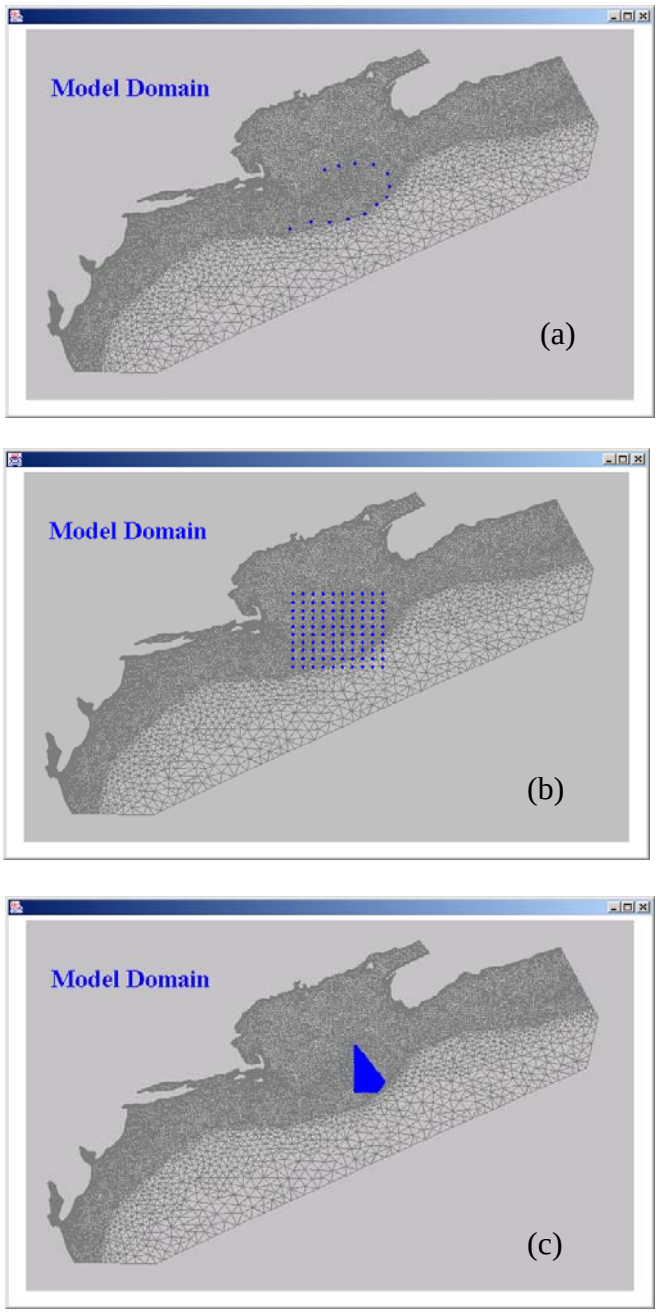

Figure 3.2 - Setup Initial Particle Locations

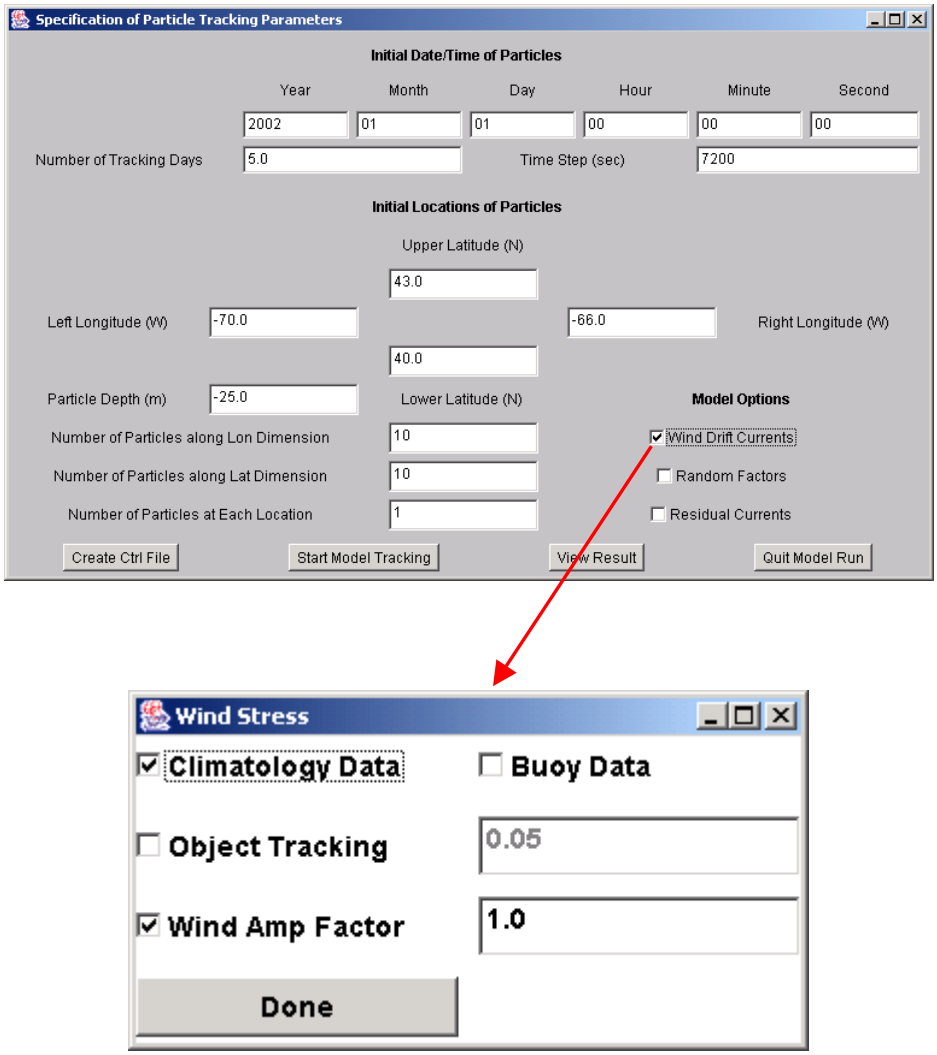

Figure 3.3 – Parameters of Wind Drift Currents

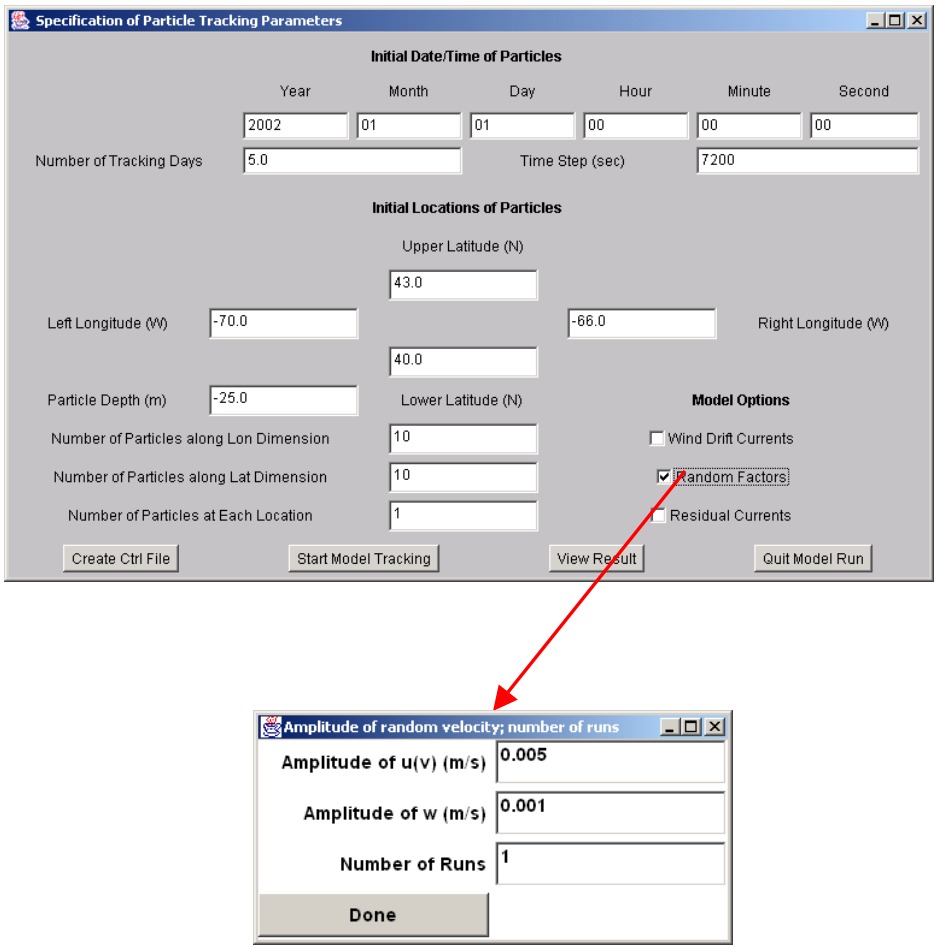

Figure 3.4 – Parameters of Velocity Components Due to Random Walks

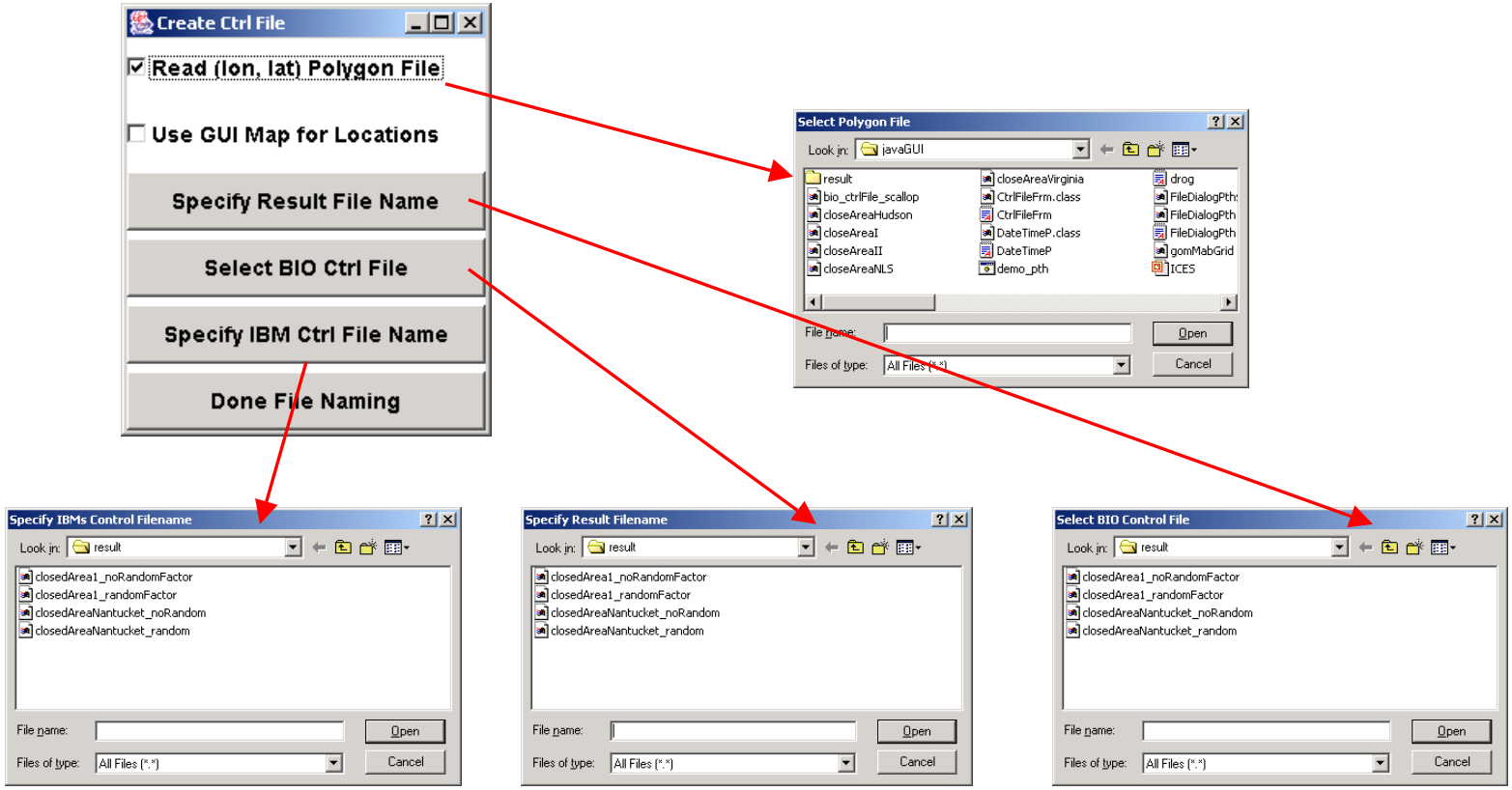

Figure 3.5 – Create File Frame and File Browser Frame

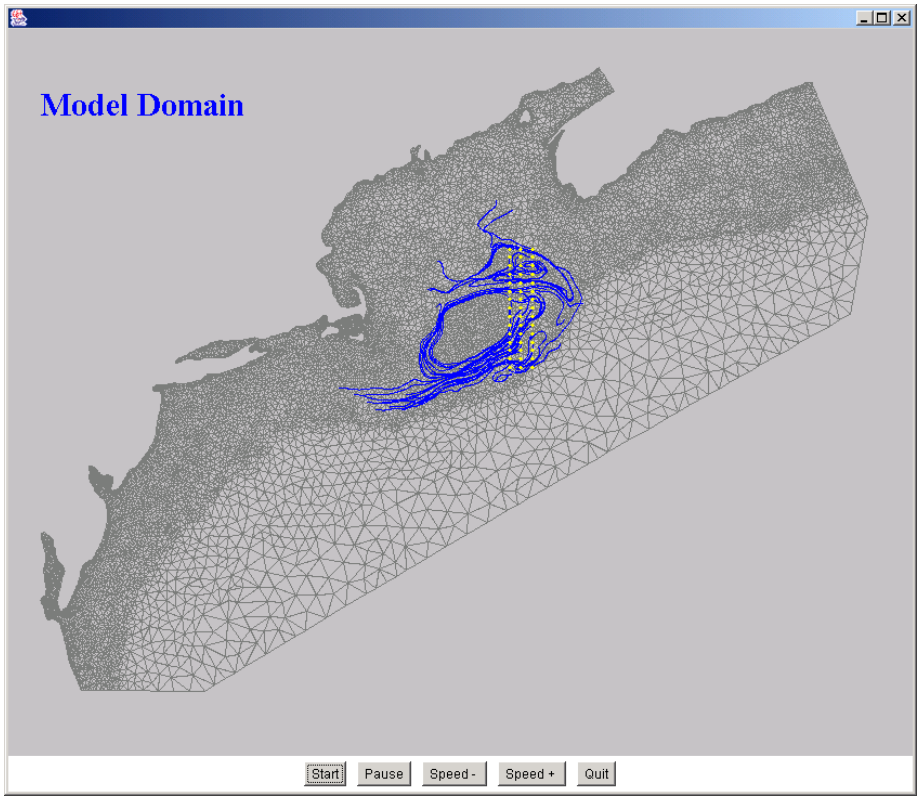

Figure 3.6 – Graphical User Interface for Displaying Particle Path

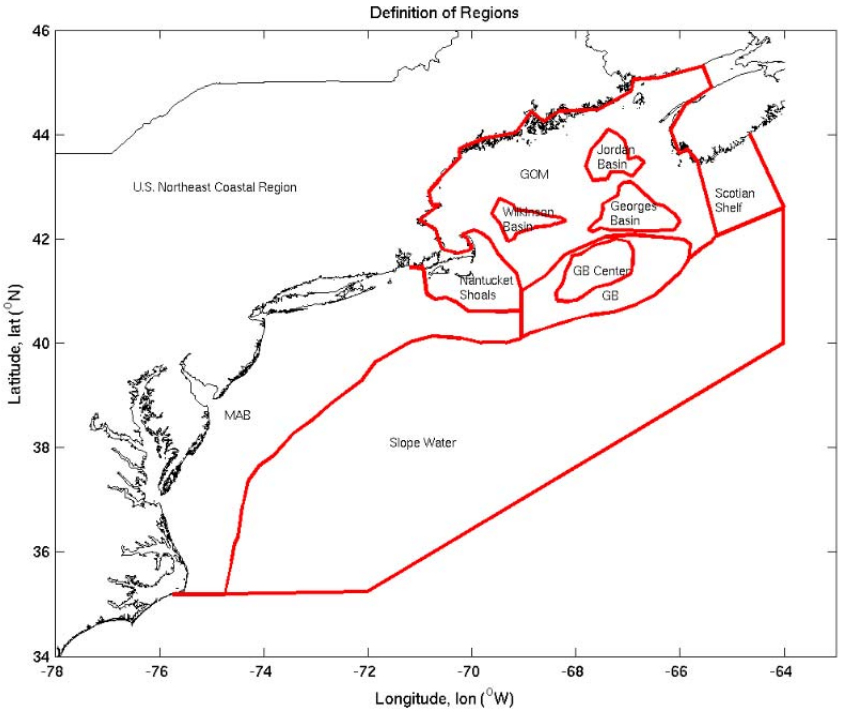

Figure 5.1 – Subregion Definitions

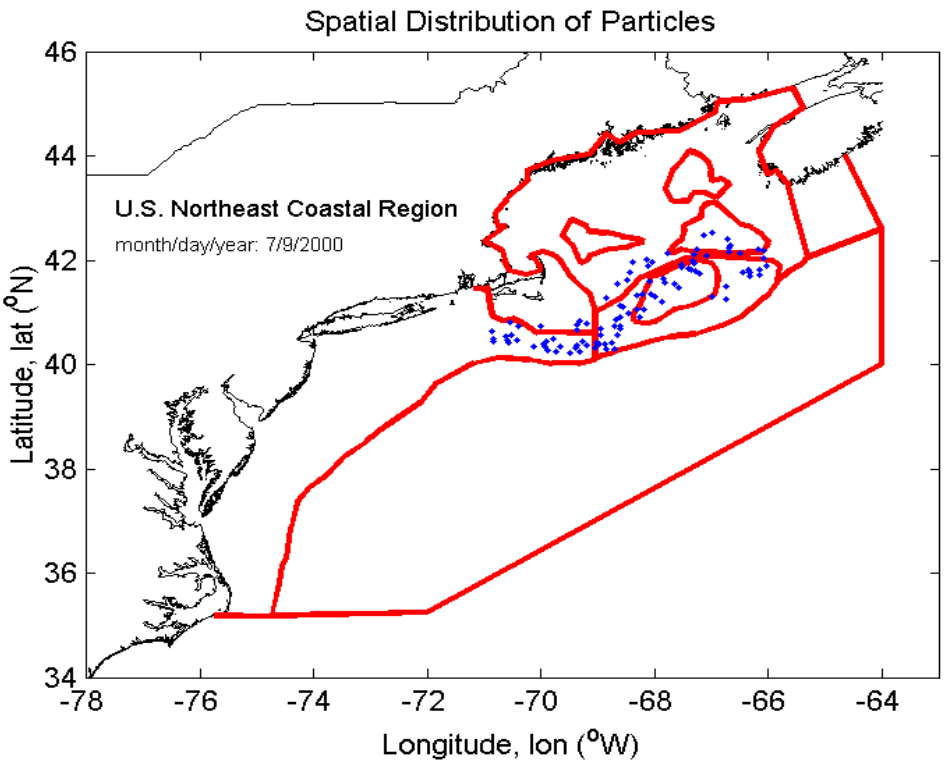

Figure 5.2 – Instantaneous Horizontal Locations of Particles

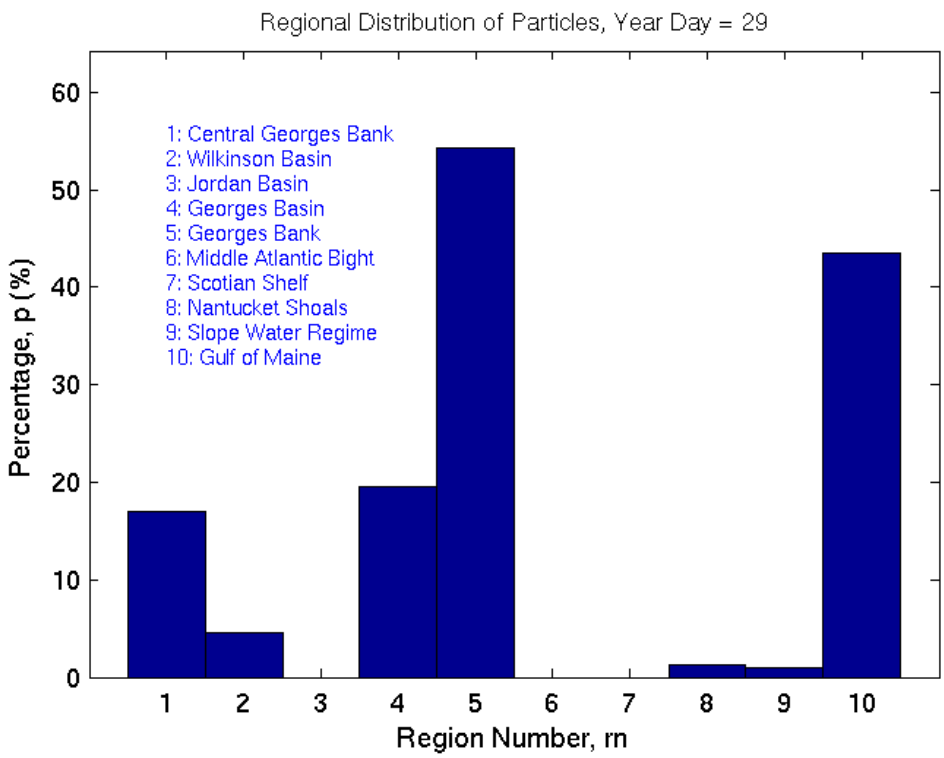

Figure 5.3 – Distribution of Particle Horizontal Locations

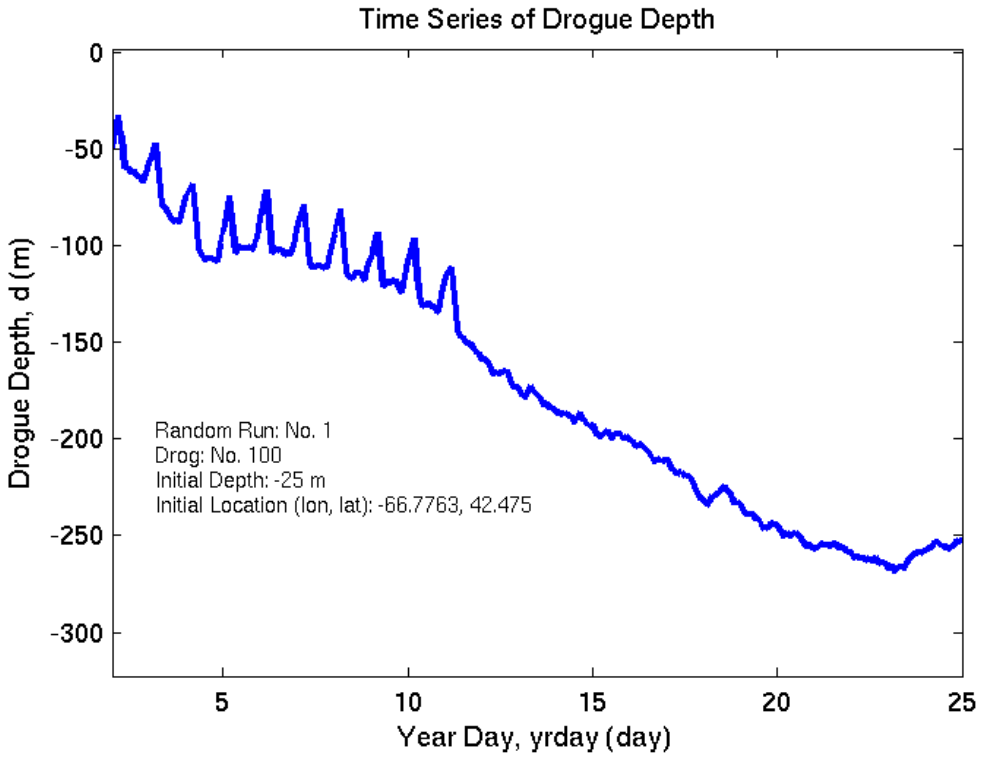

Figure 5.4 – Vertical Locations of Particle Trajectory

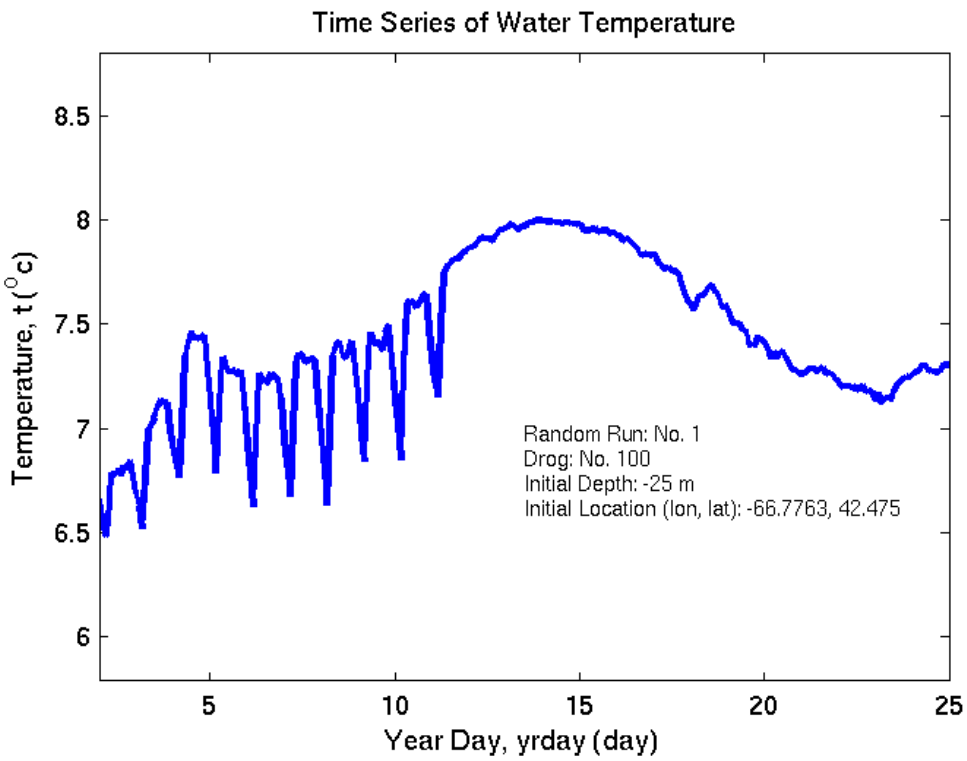

Figure 5.5 – Water Temperature along Particle Trajectory