Guide for the DieCAST Ocean Model

Yu-Heng Tseng and David E. Dietrich

Contents

Chapter 1

Introduction

This document provides the reader with the basic information necessary to carry out numerical simulations using DieCAST (Dietrich Center for Air Sea Technology) ocean model. Our main goal is to provide documentation of the DieCAST family of ocean models, a flexible and user friendly mathematical modeling system for simulating a wide variety of real and idealized ocean flows over a wide range of scales and boundary conditions. DieCAST evolved from the Sandia Ocean Modeling System developed under the Department of Eneregy sponsored Subseabed Waste Disposal Program (Dietrich *et al.*, 1987). This documentation is intended for those having a background of fundamental computational fluid dynamics concepts, some basic fluid mechanics concepts and fortran language knowledge. This documentation is illustrated by example applications and augmented by liberal comments within the model.

DieCAST is a robust, efficient, accurate family of models using fourth order accurate approximations, in a collocated (non-staggered, Arakawa 'A' grid) control volume framework, for all significant steps in the solution procedure. This includes rigorously fourth-order-accurate derivation of control volume face averages from the fundamental control volume averages of the conserved quantities (momentum, salt, heat and other quantities such as oxygen that may be needed for a particular problem).

The basic model features include incompressible, Boussinesq and hydrostatic approximations, as documented herein (nonhydrostatic versions exist); rigid lid approximation (good for slow modes that dominate ocean general circulation, but a free surface version also exists); open boundary conditions; one- and two- way grid coupling; an equation of state that describes the entire range of ocean conditions; a new surface flux boundary treatment that does not damp transients and gives rapid convergence to ensemble average annual cycle climatology (Dietrich *et al.*, 2003) and may be used in combination with synoptic wind and surface flux forcing.

Chapter 2 presents some useful links and typical results from selected applications to show model capabilities. Chapter 3 gives governing equations and the numerics. Chapter 4 tells how to obtain the code and get

started, including the basic inputs required. Chapter 5 gives simple example applications and results from adaptations available on the web. Chapter refchap:directory describes the required data inputs, thereby allowing the user to develop the front end. Alternatively, the user may take advantage of a general front end that is provided as described in Section 6, which also describes model outputs and graphics postprocessing that is also provided.

Chapter 2

Links and examples of results from real DieCAST application

DieCAST model's unique combination of rigorously 4th-order-accurate numerics and non-staggered grid results in simulation of realistic eddies, including fronts, frontal eddies, bottom density currents and other fine-scale features using much less resolution than required by other models. Here are some useful links for several applications. We will include brief introduction for each example in the near future.

2.1 DieCAST relevant websites and useful links

http://www.ssc.erc.msstate.edu/DieCAST
http://web.maths.unsw.edu.au/~ bxs/DieCAST/MANUAL/index.html
http://fluid.stanford.edu/~ yhtseng/research/DIECAST/index.html
http://www.phys.ocean.dal.ca/programs/CANDIE/

2.2 Southwest Pacific Ocean model: $1/4^\circ$ resolution

http://www.ssc.erc.msstate.edu/DieCAST/spocm/index.html
http://www.ssc.erc.msstate.edu/DieCAST/spocm/index.html#East

2.3 North Atlantic Ocean: $1/6^\circ$ resolution(Dietrich *et al.*, 2004)

http://fluid.stanford.edu/~ yhtseng/research/duo_nab/nab_animation.html

2.4 Adriatic Sea: $1/50^\circ$ resolution(Cushman-Roisin *et al.*, 2004)

http://thayer.dartmouth.edu/~ cushman/papers

2.5 Gulf of Mexico: $1/6^\circ$ and $1/12^\circ$ resolution(Dietrich, 1997)

http://www.ssc.erc.msstate.edu/DieCAST

2.6 Monterey Bay regional circulation: $1/72^\circ$ resolution(Tseng *et al.*, 2004)

http://fluid.stanford.edu/~ yhtseng/research/movies.html#Monterey

Chapter 3

Governing equations and numerical descriptions

The DieCAST family of full primitive equations (Boussinsq approximation) ocean models is documented herein. The hydrostatic, rigid-lid approximation version documented herein is freely available. Hydrostatic free surface and nonhydrostatic members are available by license agreement. The free surface version uses an efficient 'truly leapfrog' (interlaced in time) approach for the time-split barotropic 'fast mode'. The fast mode is governed by the shallow water equations, including forcing terms from the baroclinic mode. This chapter provides the governing equation and some numerical procedures.

3.1 Basic DieCAST ocean model description

DieCAST is a 'z-level model'; geopotential is its vertical coordinate. Geopotential, longitude and latitude iso-surfaces define a set of 'control volume boxes' or 'cells' in a spherical coordinate system. The conservation laws are integrated in each cell to get 'control volume' (cell averaged) equations governing cell mean values. Reynolds stress turbulence closure for subgrid-scale transports across the cell faces is parameterized by user-defined eddy viscosity and diffusivities; they are defined as:

a) for vertical cell faces, a modified Pacanowski & Philander (1981) approach specifies vertical viscosity and diffusivities based on the Richardson number.

b) for horizontal cell faces, the default is constant horizontal viscosity and diffusivities; however, the user may specify values in a three dimensional scratch array, which is then interpolated to the cell faces.

The same set of cells is used for all conservation quantities; momentum density (velocity) component cells are NOT staggered relative to pressure, temperature, salinity and other predicted scalar quantities. This avoids interpolation errors in evaluating the large Coriolis term.

3.1.1 Continuum equations

DieCAST divides the ocean domain into a large number of cubical finite 'control volumes' or 'cells'. Usually, most contain water, while others are on land or below the bottom. To reduce wasted computation in masked land areas, one may use domain decomposition with domain coupling using the robust DieCAST grid coupling approach. The conservation equations are applied (in integral form) to each individual control volume; this means that time changes are calculated from a sum of flux approximations across the six sides of the cubical control volumes plus special terms such as pressure gradient and Coriolis. The side faces are exactly vertical. The top and bottom faces are exactly horizontal. Bottom and lateral boundaries are approximated by a stairstep approximation. The sides are assumed to meet at right angles, as in Cartesian and spherical coordinates. We present the details of DieCAST hydrostatic model in this section. The governing equations have

Conservation of mass:

$$\nabla \cdot \mathbf{V} = 0 \tag{3.1}$$

Horizontal momentum equations:

$$\frac{\partial u}{\partial t} = -\nabla \cdot u\mathbf{V} + fv - \frac{1}{\rho_0}\frac{\partial p}{\partial x} + \nabla_h \cdot A_h \nabla_h u + \frac{\partial}{\partial z}(A_v \frac{\partial u}{\partial z}) \tag{3.2}$$

$$\frac{\partial v}{\partial t} = -\nabla \cdot v\mathbf{V} - fu - \frac{1}{\rho_0}\frac{\partial p}{\partial y} + \nabla_h \cdot A_h \nabla_h v + \frac{\partial}{\partial z}(A_v \frac{\partial v}{\partial z}) \tag{3.3}$$

Conservation of scalar (salt or potential temperature):

$$\frac{\partial S}{\partial t} = -\nabla \cdot S\mathbf{V} + \nabla_h \cdot K_h \nabla_h S + \frac{\partial}{\partial z}(K_v \frac{\partial S}{\partial z}) \tag{3.4}$$

Hydrostatic equation:

$$\frac{\partial p}{\partial z} = -(\rho - \overline{\rho})g \tag{3.5}$$

Equation of state:

$$\rho = \rho(S, T) \tag{3.6}$$

where u and v are the x and y components of velocity vector $\mathbf{V} = (u, v, w)$ respectively; f is the Coriolis parameter; ρ_0 is the mean density; $\overline{\rho}$ is the horizontally averaged density at depth z; p is the pressure; A_h, A_v are the horizontal and vertical eddy viscosity respectively; S is the salinity; K_h, K_v are the horizontal and vertical eddy diffusivity respectively; T is the potential temperature.

From the staggered grid arrangement shown in Figure 3.1, the control

volume and time integrated form of equation (3.2) may be written,

$$\begin{aligned}\frac{\tilde{u}^{n+1}_{i,j,k} - u^{n-1}_{i,j,k}}{\Delta t} = &- \Big(\frac{U^n_{i,j,k}u^n_{i+1/2,j,k} - U^n_{i-1,j,k}u^n_{i-1/2,j,k}}{\Delta x_j} \\ &+ \frac{V^n_{i,j,k}u^n_{i,j+1/2,k} - V^n_{i,j-1,k}u^n_{i,j-1/2,k}}{\Delta y_j} \\ &+ \frac{W^n_{i,j,k+1}u^n_{i,j,k+1/2} - W^n_{i,j,k}u^n_{i,j,k-1/2}}{\Delta z_k}\Big) \\ &+ [2\Omega_e sin(\phi_j) + u^n_{i,j,k}\tan(\frac{\phi_j}{r_e})]v^n_{i,j,k} \\ &- (\frac{\partial \tilde{p}}{\partial x})^n_{i,j,k} + dissipation\end{aligned} \tag{3.7}$$

where the pressure derivative term is

$$\begin{aligned}\frac{\partial \tilde{p}}{\partial x}^n_{i,j,k} = &\frac{7}{12}\Big(\frac{\tilde{p}_{i+1,j,k} - \tilde{p}_{i,j,k}}{\Delta x_j} + \frac{\tilde{p}_{i,j,k} - \tilde{p}_{i-1,j,k}}{\Delta x_j}\Big) \\ &- \frac{1}{12}\Big(\frac{\tilde{p}_{i+2,j,k} - \tilde{p}_{i+1,j,k}}{\Delta x_j} + \frac{\tilde{p}_{i-1,j,k} - \tilde{p}_{i-2,j,k}}{\Delta x_j}\Big)\end{aligned} \tag{3.8}$$

and the dissipation term is the viscous and quadratic bottom drag (bottom layer only) terms. In addition, a wind stress is applied to the top layer. Although standard Earth spherical coordinates are assumed, we do not include the cosine factors here (horizontal grid increments vary with latitude only and vertical grid increments with depth only). The quantity $\tilde{q}$ means the trial value of field q derived from the trial pressure at the rigid-lid. The rigid-lid pressure is hydrostatically related to the slow mode surface height.

In order to evaluate a cell face average quantity, fourth order accurate interpolation is used (see Appendix B): Then $\overline{q}_{i+1/2,j,k}$ represents the cell face quantity derived from the four nearest control volumes, for example:

$$\overline{q}_{i+1/2,j,k} = \frac{7}{12}(Q_{i,j,k} + Q_{i+1,j,k}) - \frac{1}{12}(Q_{i-1,j,k} + Q_{i+2,j,k}) \tag{3.9}$$

where $\overline{q}_{i+1/2,j,k}$ represents the cell face average quantity derived from its four nearest control volume averages on the right hand side of equation (3.9). After determining the pressure field $\tilde{p}$ from the hydrostatic equation (3.5) using a trial (or most recent) value of rigid-lid pressure, the control volume conservation equations are used to get $\tilde{u}, \tilde{v}, S, T$ at the advanced time level $(n+1)$. All terms except diffusion are evaluated from fields known at leapfrog time level n; diffusion terms are at time level $n-1$. The numerical procedure for the incompressible correction is laid out in Appendix A.

3.1.2 The DieCAST computational grid

The resolution parameters (the number of control volumes in each direction) in DieCAST are $I0$, $J0$, and $K0$. Figure 3.2 shows how they are laid out.

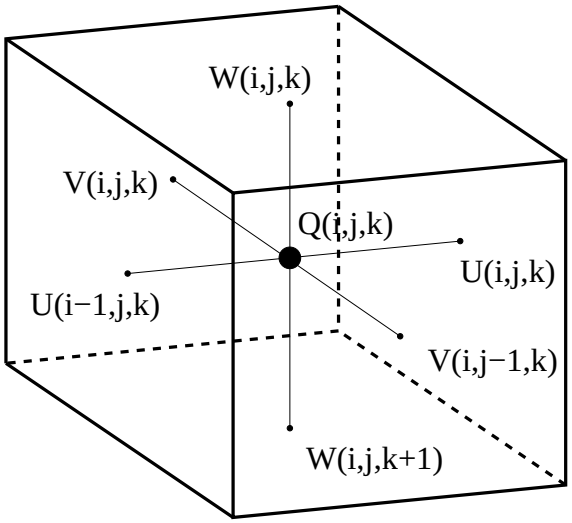


Figure 3.1: The staggered and collocated arrangement of grids. $Q(i,j,k)$ is any cell-averaged quantity, u, v, s, t, p or ρ. U, V, W are the face-averaged quantities.

This includes 'ghost zones' at $I = 1$, $I = I0$, $J = 1$ and $J = J0$, which are used for convenient specification of inflows and outflows at open boundary points (see Section 3.2.3). No ghost zones are used in the vertical direction. The conservation equations are applied only to the 'interior' control volumes that actually contain water. In the modified 'a' grid DieCAST model these control volumes are the same for all fields. The 'c' grid SOMS model, from which DieCAST evolved (Dietrich *et al.*, 1987), is similar except the control volumes for the horizontal momentum conservation are laterally staggered 1/2 grid interval from the control volumes for scalar quantities pressure, temperature and salinity.

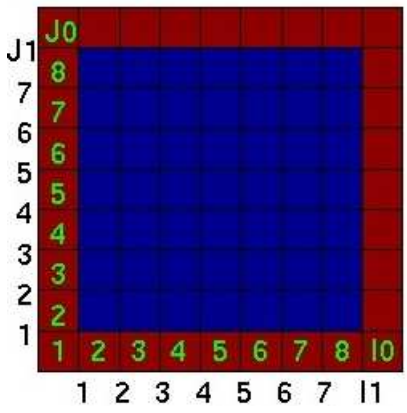


Figure 3.2: Computational grid.[1]

[1]http://web.maths.unsw.edu.au/~ bxs/DieCAST/MANUAL/introduction.html

The modeled fluid, to which the collocated-control-volume-integrated conservation laws are applied, is contained in the between $I = 2$ and $I = I0 - 1$ ($I1$ in the model variables) in the x-direction, between $J = 2$ and $J = J0 - 1$ ($J1$ in the model variables) in the y-direction, and between $K = 1$ and $K = K0 - 1$ ($K1$) in the z-direction. Thus, the number of interior control volumes is:

- $I0 - 2$ ($I2$ in model variables) in the x-direction
- $J0 - 2$ ($J2$ in model variables) in the y-direction
- $K0 - 1$ ($K1$ in model variables) in the vertical(z) direction

Thus, $K0$ is the number of layer interfaces in the vertical, where the staggered vertical advection velocity is located, and $K0-1$ ($K1$) represents the number of vertical 'control' volumes.

The following is critical to the understanding and application of DieCAST. A master mask array called 'IN' defines all boundaries. Together with the metrics arrays, this defines the region modeled. For scalar cell I, J, K filled with water (land), $IN(I, J, K) = 1(0)$. Thus, land and topography are specified by a stairstep approximation. The other mask arrays IU, IV and IW are derived from IN. All mask arrays are $integer2$ and 3-dimensional.

A row (column) of scalar ghost zones is included around the logically rectangular perimeter of the modeled region. These zones are used to store temperature, salinity, and momentum data that can be advected into the modeled region through inflow cell faces. The data is often specified from climatology, but other specifications can be used depending on application.

At interior points adjacent to boundaries, normal open-boundary fluxes are specified by an upwind approximation at the boundary. For inward flow, the aforementioned ghost zone data is advected inward. For outward flow, the interior data is advected outward. The normal velocity on the boundary is updated separately, by:

a) combining psuedo-Neuman (for outward flow) and climatological restoring effects or

b) by near-boundary data from an overlap region of a separate model grid (as in nesting).

In the former case, if the first interior point has outward directed normal velocity component, the boundary point may be nudged toward the nearest interior value on a short (synoptic) time scale rather than setting it directly to Neuman (which would imply infinite signal propagation speed).

This simple approach works well, although a more accurate scheme might be better. The normal flow on the logically rectangular boundary may be continuously relaxed toward a climatologically specified normal velocity on a longer time scale. The climatologically specified baroclinic component of the normal velocity should be consistent with a geostrophic thermal wind based on the climatological salinity and temperature data except very close

to the equator. The barotropic component is independent of this data, but the total velocity must be such that there is no net volume inflow from lateral boundaries because of the rigid-lid approximation.

3.1.3 Leapfrog time stepping

Leap-frog time-stepping uses three time-levels. Consider a differential equation

$$du/dt = F(u,t) \tag{3.10}$$

which can be written using second-order accurate centered time-differencing

$$U3 - U1 = 2 \times dt \times F(U2, t2) \tag{3.11}$$

so that $U3 = U1 + 2 \times dt \times F2$. To achieve second-order accuracy, we clearly need to obtain $U2$ at the central time level. This is done by cycling the variables so that $U1 = U2$ and $U2 = U3$ at the end of the time step. Note that the update from the old time level $U1$ leaps over the central level $U2$ to the leap-frog level $U3$. Generally, time-splitting of the solution will result and this is corrected by adding an Asselin filter (Asselin, 1972) with weight FLTW before cycling the variables so $U2 = (1 - FLTW) * U2 + FLTW * (U1 + U3)$ which mixes some of the information in from other time levels. This Aselin filter formally reduces the time-stepping to first-order accuracy (Roache & Dietrich, 1988) — albeit a generally accurate first-order, because very small FLTW is adequate to eliminate time splitting.

3.1.4 DieCAST open boundary treatment

DieCAST uses the following open boundary treatment. All boundaries are defined by cell faces that divide water-containing cells from land or external fluid. If there is external fluid outside the model domain, special open boundary treatment is required.

All model predicted field variables are internal cell averaged quantities by definition. In a given time step these model variables are first updated excluding effects of open boundary fluxes. This is automatically done by mask array factors that effectively give zero normal boundary fluxes on all open boundary cell faces, including islands and seamounts as well as the logically rectangular domain boundaries. The mask arrays are defined by the model preprocessor.

Full control of (and responsibility for) open boundary conditions is left to the user. Such control makes DieCAST general, efficient and relocatable but requires some expertise and care by the user as with any general ocean model.

The DieCAST model code includes open boundary effects along all lateral boundaries that fall on its logically rectangular grid boundaries. The open

boundary coding is designed for efficient use of vector and massively parallel computers, as is the rest of DieCAST.

DieCAST open boundary conditions are specified or derived fluxes of all transported quantities into the adjacent-to-boundary control volumes, as required at open boundaries Dietrich *et al.* (1987). This approach is simple and robust. For example, in nesting a one minute (1/120°) resolution DieCAST Santa Barbara Channel application inside a coarser five minute (1/12°) California Current 'BIG picture' application (developed in collaboration with Drs. Bob Haney and Bob Hale), using no sponge layer, very little boundary noise is generated and the fine grid solution is consistent with the coarse grid but richer in detail. We have extended that basic approach to a two-way-coupled duo grid North Atlantic Ocean adaptation of DieCAST (Dietrich *et al.*, 2004).

DieCAST open boundary conditions and nested grid coupling are based on upwind approximations. All open boundary conditions are based on boundary fluxes. Users can refer to Tseng *et al.* (2004) for more detailed one or two way couplings. Point sources, such as from rivers smaller than one grid point, are treated by specifying salinity (zero), heat, momentum and volume inflows at specified source locations; this is done through vertical velocity and fluxes specified through a 'semi-porous' rigid-lid but may be generalized for deeper river inflows. 'e-p' (evaporation - precipitation) is similarly specified by vertical velocity relative to the modeled ocean surface (semi-porous rigid-lid).

3.2 Basic program sequence

The basic computation procedure during a model time step is as follows.

The first step is to calculate the local cell averaged density, from potential temperature, salinity and pressure-depth. This uses an equation-of-state provided by Dan Wright. Then, starting with a trial "surface" pressure (or hydrostatically equivalent surface height) at the top of the surface layer of cells, the hydrostatic relation is integrated analytically to get face-averaged pressures at all other top and bottom cell faces. These are then used to get cell-averaged pressures within the all cells. Next, the latter are used to get face-averaged pressures at the four lateral faces of all cells; this gives the pressure force in the cell-averaged momentum equations. These steps are all calculated with fourth-order-accuracy.

After partially updating horizontal velocity by adding the divergence of momentum flux plus diffusion fluxes to the pressure force effects, a top layer pressure adjustment is calculated such that, when added to the guessed pressure, the divergence of the vertically integrated horizontal velocity is zero (or, in the case of net specified e-p, a small specified value). This (semi-porous) rigid-lid pressure (hydrostatically equivalent to slow mode surface height) correction is governed by a Poisson equation for the top layer pressure ad-

justment; one must be careful that the boundary conditions do not violate global incompressibility — there must be no net inflow in order for an incompressible velocity field to exist and for the Poisson equation for the top layer pressure adjustment to be well posed. After the face averaged velocities are adjusted according to the rigid-lid pressure adjustment, their CHANGES are interpolated back to give CHANGES of the cell averages, again using rigorously fourth-order-accurate interpolations. The algebraic details of this procedure are given in Appendices.

Vertical velocity is derived simply by vertical integration of the divergence of the horizontal velocity, as per the incompressibility approximation to the mass conservation equation.

The numerical details of this straightforward, rigorously fourth-order-accurate procedure are also documented by extensive comments among the lines of model code, and by Dietrich (1997). This formal accuracy gives an accurate, robust model. Only the barotropic mode pressure adjustment is second-order-accurate; yet, it is quite accurate due to the staggered grid arrangement between barotropic mode pressure and velocity.

The above procedure includes the divergence of the barotropic mode of advection velocity, e.g., as implied by non-zero e-p in the framework of surface boundary conditions that do not require transient-damping nudging toward climatological data (Dietrich *et al.*, 2003). This new approach is implemented in recent DieCAST model adaptations including the North Atlantic Ocean (Dietrich *et al.*, 2004).

3.3 Model features

This documented DieCAST version uses standard Boussinesq, hydrostatic and rigid-lid approximations. It includes:

1. full spherical metrics with variable grid increments;

2. a standard nonlinear equation of state relating in-situ density to potential temperature, salinity and pressure-depth;

3. 3-d turbulent vertical diffusivity arrays; in the SOMS model from which DieCAST evolved, these were determined by computation- and storage-intensive Mellor-Yamada level 2.5 turbulence closure (Dietrich *et al.*, 1987); the present default is the Pacanowski & Philander (1981) approach; the model user may apply any alternate preferred scheme.

4. 3-d turbulent horizontal diffusivity arrays (no diffusive turbulence parameterization is deemed effective for horizontal Reynolds stress representation in general, but these arrays are available for numerical and weakly physical based usages).

For efficiency in general circulation (slow mode) applications, DieCAST uses a rigid-lid approximation.

3.4 Advanced features

3.4.1 Free surface

An explicit time-split free surface version exists (available under license) that integrates interlaced leap-frog shallow water equations using multiple 'fast mode' time steps each 'slow mode' time step; this version addresses fast free surface modes (tides, wind surge events, etc) and is efficient, accurate and robust, including vertically averaged baroclinic pressure gradient and momentum flux forcings. The motivation for this is the first derivative coupling of different fields, similar to the motivation for spatial grids.

Incidentally, the shallow water equations are a one-layer application of sigma coordinates. They are useful for the 'fast modes' of ocean dynamics. Generally, these fast modes have only a very minor effect on the ocean 'slow modes' that dominate ocean material trajectories and general circulation, other than perhaps turbulent dissipation from breaking wave effects in localized coastal regions. The extent to which such localized dissipation affects the general circulation, and thus climate, is not well understood, but short term local effects are quite small.

Thus, a general circulation model may run independently, creating vertically averaged baroclinic pressure gradients and fluctuation correlations that may be used by a more specialized free surface 'fast mode' model, while ignoring at least short term feedbacks from the barotropic mode (shallow water equations solution) to the baroclinic mode. Thus, models using the rigid-lid approximation can be quite useful, as proven by their many demonstrated successes in general circulation studies.

3.4.2 Nesting and grid coupling

See Dietrich *et al.* (2004) for basic methodology and results from application to North Atlantic Ocean. Model details available under licensing agreement.

3.4.3 Non-hydrostatic

For basic methodology, see Dietrich & Lin (2002) and Tseng & Ferziger (2003) for the numerical procedure. Model details available under licensing agreement.

3.4.4 Data assimilation

Under development.

Chapter 4

Getting started

running sample programs on your computer

Basic user instruction will be introduced in this chapter. We have two step-by-step examples of running DieCAST in next chapter. First time users can run the code first (see next chapter) and then come back to this section for more I/O details.

4.1 Preprocess and data input

DieCAST reads input data from subdirectory **PREP/DATA**. The code **PREP/prep.f** creates the following files for input to DieCAST: a two-dimensional topography array **KB(I,J)** giving the number of layers that are water- occupied at a given horizontal grid point (I,J), and other masking and geometry arrays; files for initial temperature and salinity (not needed when restarting from previous model results); surface wind forcing and restoring data files; open lateral boundary files; and matrix data for the EVP surface pressure solver. Input data to **prep.f** comes from the subdirectory **DATA**, and is generated by programs **metgen.f**, **inmets.f** and **indata.f** in that directory which read and process global climatological data to the model grid; and most of **prep.f** output data goes there also, to be read by the DieCAST model.

One may write alternate program(s) to replace **prep.f**, or to create the depth, intial, and boundary data that **prep.f** reads and processes for input to DieCAST. Most input data is read by the **INITxx** subroutines of DieCAST.

DieCAST writes output fields to three output files, one in each of three subdirectories, as follows:

XYPLOTS/DATA: horizontal plots

XZPLOTS/DATA: vertical east-west cross-sections

YZPLOTS/DATA: meridional cross-sections

In addition, for optional (when **LMOVI=1**) computer-generated movies of DieCAST output, data is written to the **MOVIE** subdirectory:

MOVIE/DATA: horizontal field animation (pressure contours and tracer streaklines)

To minimize storage, output fields are scaled and written in binary **integer*2** form and grid locations are derived from the small basic metrics data sets rather than combined with the data which would be hugely wasteful of storage. Scale factors are used to recover original field values in graphics postprocessors.

The first step in running DieCAST is to specify input data for the program **prep.f** (which resides in subdirectory **PREP**) which controls DieCAST. This includes primary scalar parameters, defined in its first subroutine (**BLOCK DATA NISHAL**). Bathymetry, wind stress, temperature and salinity surface forcing and initialization data are derived from program files under subdirectory **DATA**, often from DBDB5(Digital Bathymetric Data Base at 5 minute resolution) global bathymetry data sets, Hellerman or COADS (Comprehensive Ocean-Atmosphere Data Set) winds and Levitus or Yashayaev annual cycle temperature and salinity climatology.

Open lateral boundary data may also be specified by **prep.f**. A common practice is to nudge model surface temperature(Haney, 1971) and salinity toward data from Levitus climatology; the model uses a better approach(Dietrich *et al.*, 2003). Open lateral boundary data is stored in the aforementioned ghost zones, and often comes from Levitus climatology.

Next, compile and run prep.f, which is documented by extensive internal comments that explain its variables and the sequential steps used to generate initial and boundary condition data. After running **prep.f**, one simply compiles and runs the DieCAST model in the overlying directory.

4.2 Graphics output

Graphics output, mostly in the form of contour plot data for various two-dimensional slices of the model fields at user-specified time intervals (from **prep.f**), requires four main steps after running DieCAST:

1. Go to the subdirectory of interest (for example, for horizontal plots, use subdirectory **XYPLOTS**).

2. Compile and run **plot0.f**. This creates: **CONVEC**, a file of contour and vector line segments; geometry files **GRID** and **OBSTACLES**; and **LINES**, a text file which gives the dimensions required to run the graphics processor program called **plots.f**.

3. The program **plots.f** is the only program in this sequence that requires linking to a graphics library. Everything else is totally portable standard Fortran. Even **plots.f** is totally portable if one foregoes the viewing of plots on the computer monitor, because **plots.f** has an option to make Postscript graphics data files only (with no screen previewing)

which can be printed on any Postscript compatable printer or viewed on the screen using **gv** or **ghostview freeware**. The screen previewing is a very useful feature that allows one to 'screen out' the plots that one desires to make into hardcopies.

NOTE: Make sure the dimension parameters in **plots.f** are at least as large as the specified values in the file called **LINES**.

---

4. Run the **plots.f** executable.

   With openGL screen previewing, prompts will appear in the 'plots' control window. First, the user will be asked to choose one of three options: (a) just viewing the plots on the monitor; (b) using the monitor as a previewing device to decide which plots will stored in a Postscript file; c) just making a Postscript file that includes all plots.

The Postscript file is a text file containing sequence of Postscript graphics language commands. The commands are automatically written by the executable '**plots**' to a Postscript command file **ps.dat**. To see the plots on the screen, use freeware **gv** or **ghotstview**. To get hard copies, print the file **ps.dat** on a Postscript compatable printer. A special control character is written at the beginning of the Postscript file to tell the printer that what follows is a Postscript file. Although Postscript files can be very large, Postscript is a widely supported graphics language with many useful features. Postscript files can be greatly compressed using the Unix **gzip** command.

The DieCAST model, its preprocessors and graphics postprocessors are documented by many comments among the lines of code. Appendix D includes listings of these programs, subroutines and plots of sample outputs. Appendix E gives an alphabetical listing of the variables and their meanings.

Chapter 5

Tutorials

This section describes how to use the DieCAST ocean model using two examples and provides step-by-step tutorials. Those include a numerical simulation of island wakes in the ocean (Dietrich *et al.*, 1996) and a three dimensional gravity plume on a continental slope(Tseng & Ferziger, 2004). The best way to familiarize yourself with the model is to run these two cases directly. Information on how to obtain, compile and run the code is found here.

5.1 Obtain the code

You can download the two examples from the following website *http://fluid.stanford.edu/~ yhtseng/research/DIECAST/* As a courtesy we ask you to send email to us at yhtseng@stanfordalumni.org to enable use to keep track of who's using the model in what application.

5.2 Numerical simulation of island wakes

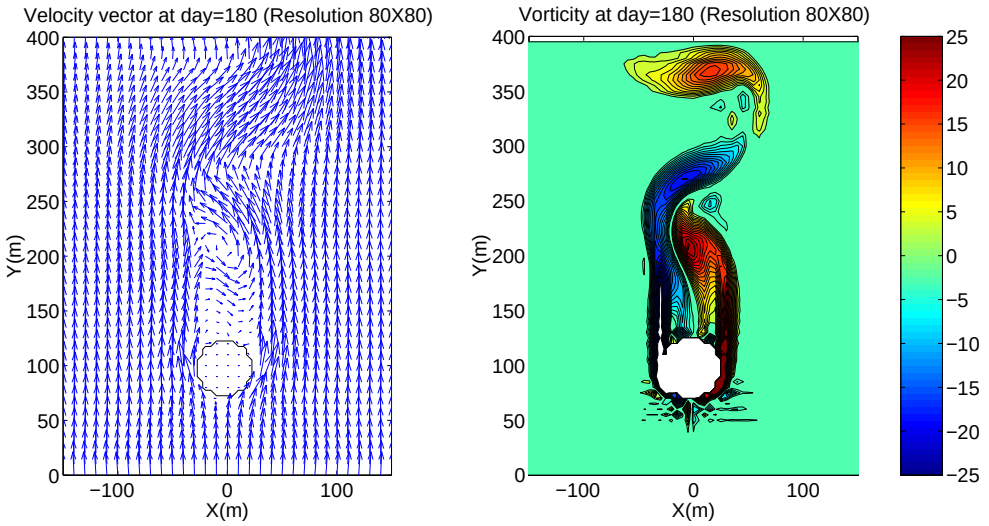
5.2.1 Description

The two dimensional, hydrostatic DieCAST ocean model simulates the idealized oceanic flow around small islands patterned after Barbados, W. I. (Dietrich *et al.*, 1996). The effects of vortex shedding and wake patterns can be examed by changing the Reynolds number. The velocity and vorticity fields are illustrated in Figure 5.1 at day 180.

5.2.2 Running the code

1. unzip the **ILE2D.tar.gz** (UNIX) or **ILE.zip** (WINDOWS) file. It will automatically create a directory called ILE2D by typing

   tar -xzvf ILE2D.tar.gz (UNIX)



(a) (b)

Figure 5.1: (a) The velocity field at day 180 for resolution 80×80. (b) The vorticity field at day 180 for resolution 80×80.

2. make sure there are two directories under ILE2D. One is **ILE2D/PREP/** and the other is **ILE2D/XYPLOTS/**.

3. run the preprocesser program in **ILE2D/PREP/** directory by typing:

   % cd PREP

   % f77 prep.f -o prep (or any Fortran compiler you are using)

   % ./prep

4. if the program runs well, it will generate a file call **KBVIEW** in **ILE2D/PREP/** and four necessary data under directory **PREP/DATA/**. The **KBVIEW** provides the grid information and resolution etc.

5. run the main code in **ILE2D/** by typing

   % f77 ile.f -o ile (You can add any optimization flag if needed)

   % ./ile > output.txt

6. the code should be done within 5 mins for a 180 days simulation in modern computers. You can compare your **output.txt** with the example run under directory **ILE2D/VERI/**.

5.2.3 Viewing the results

There are two way to view the results quickly. If you have MATLAB, you can jump to section 5.2.3. However, you can also view the results immediately on

your computers using **gv** or **ghostview** (see Chapter 4 for the information).

Using the existing plotting program and gv

You can view the results directly under directory **ILE2D/XYPLOTS/** by typing the following

% cd XYPLOTS
% f77 plot0.f -o plot0
% ./plot0
% f77 plots.f -o plots
% ./plots

The above steps will generate a large ps file called **ps.dat**. Just use your ghostview to read it by typing

% gv ps.dat or **%ghostview ps.dat**

You should be able to see the results immediately.

Using MATLAB commercial software

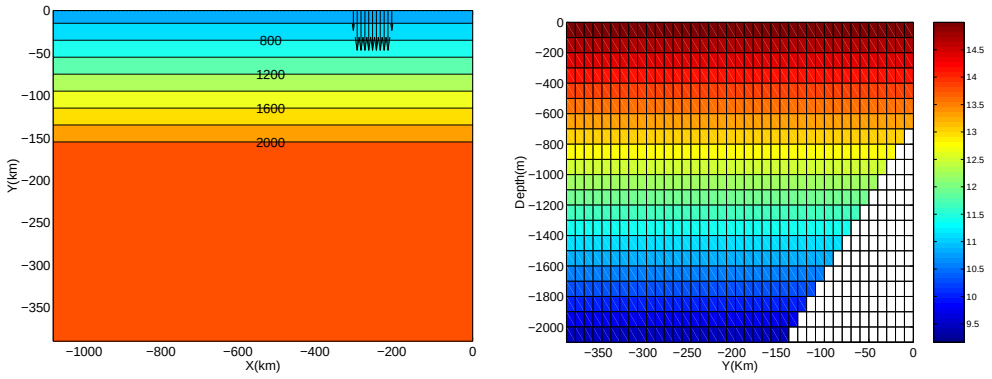
You can download the MATLAB script from the following website *http://fluid.stanford.edu/~yhtseng/research/DIECAST/matlab.html*. The file includes all necessary files and directories you need. You will see file name starting with 'plot_data_*.m' and a subdirectory 'matlab/'. For island wakes problem, just run the file 'plot_data_ile.m' and you will get the results immediately.

5.3 Three dimensional density current over a continetal slope

5.3.1 Description

The DieCAST model is applied to the DOME (Dynamics of Overflow Mixing and Entrainment) experiment. A $1100km$ (longitude) by $400km$ (latitude) region is modeled (Figure 5.2). The uniform vertical and horizontal resolution is 100 m and 10 km, respectively. The northernmost latitude is 500 m deep. A 1% shelf slope brings it to maximum depth $2200m$ at the latitude 17/40 of the total width to a closed southern boundary (Figure 5.2(b)). The depth is uniformly $2200m$ further south. This domain is the same as the high resolution experiment Z4 in Ezer & Mellor (2004). Inflow is imposed from the northern boundary, patterned after the Denmark Strait (Figure 5.2(a)).

The northern inflow is bottom concentrated and geostrophically balanced. The bottom layer (at depth 500-$600m$) inflow temperature increases linearly from $5.3°C$ on its west side to $9.9°C$ on its east side. The top layer inflow temperature increases linearly from $13°C$ on its west side to $15°C$ on its east side. The inflow is bottom-concentrated, with a total of



(a) (b)

Figure 5.2: (a) The horizontal domain and northern inflow velocity vector at the bottom. (b) Illustration of latitudinal/vertical grid and initial temperature stratification.

$4.98Sv(1\ Sverdrup = 10^6 m^3/s)$ inflow. Initial conditions are identical to the DOME conditions used by Ezer & Mellor (2004); Ezer (2004). The eastern and southern boundaries are closed. The western boundary is open, using upwind-based outflow conditions modified to give no net inflow into the modeled region. For simplicity, we use constant horizontal viscosity (ν) and diffusivity (κ) of 50 and 10 m^2/s, respectively, and a modified Pacanowski & Philander (1981) Richardson-number based vertical viscosity and diffusivity (background values of 0.1 cm^2/s). These conditions give typical cell Reynolds number $O(100)$. Resolutions is 10 km horizontal resolution with 100 m vertical resolution. The users can double/change that resolution using the same boundary conditions, viscosity and diffusivity.

Figure 5.3 shows bottom boundary temperature anomaly and velocity field at day 15, 30 and 40. The westward and downslope propagation of the dense bottom plume is clearly seen and very similar to the observations reported in previous studies (Girton & Sanford, 2003; Ezer & Mellor, 2004).

5.3.2 Running the code

1. unzip the **DENSITY.tar.gz** (UNIX) or **DENSITY.zip** (WINDOWS) file. It will automatically create a directory called DENSITY_CURRENT by typing

   tar -xzvf DENSITY.tar.gz (UNIX)

2. make sure you have the following directories for the output

   DENSITY/PREP/DATA/

   DENSITY/XYPLOTS/

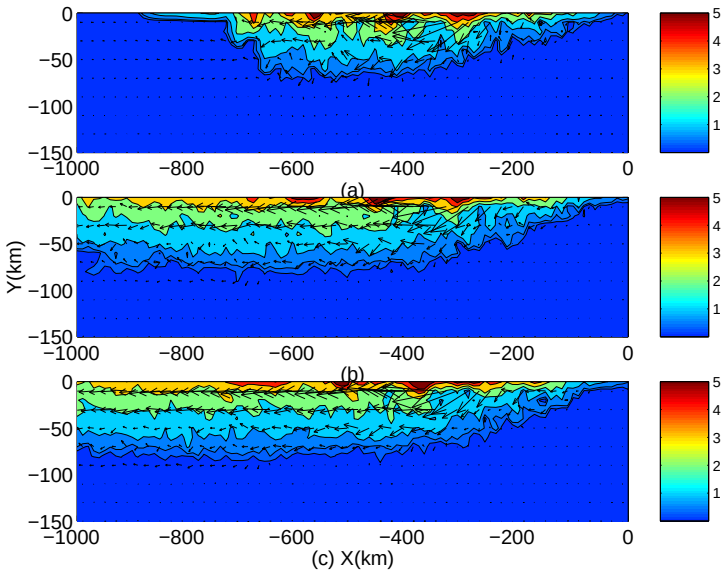


Figure 5.3: Bottom boundary temperature anomaly and velocity field at three different days ((a) Day 15; (b) Day 30; (c) Day 40).

DENSITY/XZPLOTS/

DENSITY/YZPLOTS/

DENSITY/MOVIE/)

If you don't have it, create them by typing **% mkdir PREP/DATA % mkdir XYPLOTS XZPLOTS YZPLOTS MOVIE**

3. Generate the grid and mesh in **DENSITY/PREP/DATA** directory **% cd PREP/DATA % f77 metgen.f -o metgen % ./metgen % f77 indata.f -o indata % ./indata**

4. run the preprocesser program in **DENSITY/PREP/** directory by typing:

   % cd ../

   % f77 prep.f -o prep (or any Fortran compiler you are using)

   % ./prep

5. if the program runs well, it will generate a file call **KBVIEW** in **DENSITY/PREP/** The **KBVIEW** provides the grid information and resolution etc.

6. run the main code in **DENSITY/** by typing

   % f77 newpg.f -o newpg (You can add any optimization flag if needed)

   % ./newpg > output.txt

5.3.3 Viewing the results

There are two way to view the results quickly. See section 5.2.3 for the description. In **DENSITY/XYPLOTS/**, there are three different plotting files. Replace the above **plot0.f** file with each of them. (**plot0_all.f** retrieves all information from DATA; **plot0_bottom.f** retrieves bottom boundary layer from DATA; **plot0_surf.f** retrieves surface information from DATA). You may also download the matlab scripts to view the results from the following website
http://fluid.stanford.edu/~ yhtseng/research/DIECAST/matlab.html.

Just run 'plot_data_den_XY.m', 'plot_data_den_XZ.m', 'plot_data_den_YZ.m', for different plotting section.

Chapter 6

DieCAST model: directory structure, input data (preprocessor), output data, plotting and basic parameters

Filenames (paths) given below are given relative to the user's emplacement of the tarfile for the particular application. Typically, we put them under a descriptive directory name such as '**GOM5**' for our Gulf of Mexico DieCAST model adaptation having 5 minute resolution. The directory substructure and filenames are created simply by typing '**tar xvf tarname**' where **tarname** is the name of the particular application's tarfile. Usually, the resulting directory structure will include files in 'gzipped' (compressed) form, so one must type '**gunzip fn.gz**' to get files **fn** before using them.

6.1 The model preprocessor

The model preprocessor generates its input data. It is divided into four parts:

6.1.1 PREP/DATA/metgen.f

A simple program that generates the grid and associated metrics (variable DY and DZ, with DX tied to a specified constant longitudinal spherical coordinate ANGULAR increment in degrees longitude).

6.1.2 PREP/DATA/inmets.f

6.1.3 PREP/DATA/indata.f

This program interpolates standard data sets to grid specified by **metgen.f** (basically, etopo5 for bathymetry; COADS or Hellerman for winds; and Levitus annual cycle climatology for temperature and salinity).

6.1.4 PREP/prep.f

It preprocesses elliptic surface pressure solver information, specifies basic application control parameters (such as background diffusivities, time step increment and number of time steps to be executed, graphics control, etc), eliminates unstable stratification zones in model's interpolated Levitus climatology and may specify open boundary information (such as geostrophically derived inflows, barotropic mode inflows).

6.2 The model postprocessor

The model statistics data and snapshot field data outputs and graphics plotting software are in three main snapshot subdirectories: for horizontal and vertical cross-sections, and an animation subdirectory. To save storage, all output field data is scaled to $integer * 2$ format. The graphics postprocessor reads this scaled integer data and associated scaling parameters to recover the real model data and generate Postscript graphics contour and vector plot files. Plots of the model output may be viewed by printing these files on a Postscript compatible printer and/or viewing them on the screen using ghostview or gv freeware.

The main model output subroutines (**XYPLOTS**, **XZPLOTS** and **YZPLOTS** and **MOVIE**). write model output data into subdirectory files:
XYPLOTS/DATA
XZPLOTS/DATA
YZPLOTS/DATA
MOVIE/DATA.

Graphics postprocessor software that reads and makes plots from those **PLOTS/DATA** and **MOVIE/DATA** files also resides in those subdirectories (**XYPLOTS**, **XZPLOTS** and **YZPLOTS** and **MOVIE**).

Previewing of the data on the screen using open gl graphics may be done to 'screen out' the plots to be saved in Postscript plot files. Animation software using open gl graphics is also available. These more specialized tools are available on request.

The user may develop/apply special outputs, graphics processing and analysis tools (e.g., using Matlab software). Matlab scripts are available for downloading particular output files (see Chapter 5). It may be useful to start from such special outputs and graphics software.

Model output also includes a file **TR** in the model main directory that gives basic model diagnostics more often during run time. The user may add to those diagnostics and/or remove some of them.

6.3 The model program

The main directory contains the model, '**name.f**', where the generic **name** is often given a specific name or acronym describing the application to which it has been adapted (e.g., **gom.f** for the Gulf of Mexico).

The DieCAST model code includes open boundary effects along all lateral boundaries that fall on its logically rectangular grid boundaries. The open boundary coding is designed for efficient use of vector and massively parallel computers, as is the rest of DieCAST. However, for discrete (sub-grid scale) ports such as from rivers, special treatment coding is available (we included over 100 discrete river ports in DieCAST application to Doubtful Sound in Fiordland National Park, South Island, New Zealand).

DieCAST open boundary conditions are specified or derived fluxes of all transported quantities into the adjacent-to-boundary control volumes, as required at open boundaries (Dietrich *et al.*, 1987). This approach is simple and robust. For example, in nesting a one minute (1/120°) resolution DieCAST Santa Barbara Channel application inside a coarser five minute (1/12°) California Current application (developed in collaboration with Drs. Bob Haney and Bob Hale), using no sponge layer, very little boundary noise is generated and the fine grid solution is consistent with the coarse grid but richer in detail. We have extended that basic approach to a two-way-coupled duo grid North Atlantic Ocean adaptation of DieCAST (Dietrich *et al.*, 2004).

In regions where watermass transformation effects are small, a single density variable and simplified equation of state may replace the salinity and potential temperature variables and more complex E.O.S. of the basic model.

Appendix A

Incompressibility

The numerical procedure for the incompressible correction is summarized in the following:

1. Guess the trial surface pressure $\tilde{p}_s^n$ from the previous time step ($\tilde{p}_s^n = p_s^{n-1}$) and integrate the hydrostatic equation (3.5) to get the intermediate pressure field $\tilde{p}^n$ over the whole domain.

2. Update the trial integral average $\tilde{u}^{n+1}$,$\tilde{v}^{n+1}$ in the control volume using equation (3.7) and the analogous control volume equation for v.

3. Interpolate $\tilde{u}^{n+1}, \tilde{v}^{n+1}$ to $\tilde{U}^{n+1}_{i,j,k}, \tilde{V}^{n+1}_{i,j,k}$ at the cell face using 4^{th} order interpolation from equation (3.9).

4. Adjust the rigid-lid pressure such that the divergence of the vertically integrated horizontal velocity matches the specified value(near zero, but small precipitation/evaporation and river sources are included to give non-zero W at the rigid lid which is slightly porous to fresh water). This procedure uses mass conservation and the detailed procedure is listed as follows.

   - Set the final pressure $p^n = \tilde{p}^n + \Delta\tilde{p}$ where $\Delta\tilde{p}$ is due to the change of rigid-lid pressure and thus independent of depth. $\tilde{p}^n$ is derived hydrostatically from trial rigid-lid pressure in the first step. Then the final velocities can be written as

   $$U^{n+1} = \tilde{U}^{n+1} + \Delta\tilde{U} \tag{A.1}$$
   $$V^{n+1} = \tilde{V}^{n+1} + \Delta\tilde{V}, \tag{A.2}$$

   where $\Delta\tilde{U} = -\Delta t\frac{\partial\Delta\tilde{p}}{\partial x}$ and $\Delta\tilde{V} = -\Delta t\frac{\partial\Delta\tilde{p}}{\partial y}$.
   - The final velocity field satisfies mass conservation from equation (3.1) so that the vertically-integrated horizontal velocity can be determined by vertical velocity at the top and bottom:

   $$\int_0^D (\frac{\partial U^{n+1}}{\partial x} + \frac{\partial V^{n+1}}{\partial y})dz = W^{n+1}(0) - W^{n+1}(D) = 0 \tag{A.3}$$

- Substitute the final velocity integration in terms of trial velocity and the correction in equations (A.1) and (A.2), thus deriving a 2-D Poisson equation for rigid-lid pressure adjustment. Solve the Poisson equation (A.4) using an efficient EVP elliptic solver Roache (1995).

$$\int_0^D (\frac{\partial \tilde{U}^{n+1}}{\partial x} + \frac{\partial \tilde{V}^{n+1}}{\partial y})dz = \int_0^D -(\frac{\partial \Delta\tilde{U}}{\partial x} + \frac{\partial \Delta\tilde{V}}{\partial y})dz \qquad \text{(A.4)}$$

5. The velocity U^{n+1}, V^{n+1} is corrected from the pressure adjustment $\Delta\tilde{p}$ in the last step.

6. Finally, interpolate cell average changes $\Delta\tilde{u}, \Delta\tilde{v}$ from $\Delta\tilde{U}$,$\Delta\tilde{V}$ using 4^{th} order interpolation, then $u^{n+1} = \tilde{u}^{n+1} + \Delta\tilde{u}$ and $v^{n+1} = \tilde{v}^{n+1} + \Delta\tilde{v}$.

Appendix B

Fourth-order accurate interpolation from nearby control volume averages to a cell face

For simplicity, we demonstrate the interpolation scheme in the x-direction only. The same analysis can be applied to the other directions. At a given (y, z), let any variable q (e.g. temperature) be a third-degree polynomial:

$$q(x, y, z) = C_0(y, z) + C_1(y, z)x + C_2(y, z)x^2 + C_3(y, z)x^3 \tag{B.1}$$

We conveniently assign $x_{i+1/2} = 0$, see Figure (B.1) with no loss of generality, so $q(0, y, z) = q(x_{i+1/2}, y, z) = C_0(y, z)$. We can define the quantities q at a time level.

$$q_{i+2}(y,z) = \int_{x_{i+\frac{3}{2}}}^{x_{i+\frac{5}{2}}} q(x,y,z)dx$$
$$= C_0(y,z)x|_{x_{i+\frac{3}{2}}}^{x_{i+\frac{5}{2}}} + \frac{1}{2}C_1(y,z)x^2|_{x_{i+\frac{3}{2}}}^{x_{i+\frac{5}{2}}} + \frac{1}{3}C_2(y,z)x^3|_{x_{i+\frac{3}{2}}}^{x_{i+\frac{5}{2}}} + \frac{1}{4}C_3(y,z)x^4|_{x_{i+\frac{3}{2}}}^{x_{i+\frac{5}{2}}} \tag{B.2}$$

$$q_{i+1}(y,z) = \int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{3}{2}}} q(x,y,z)dx$$
$$= C_0(y,z)x|_{x_{i+\frac{1}{2}}}^{x_{i+\frac{3}{2}}} + \frac{1}{2}C_1(y,z)x^2|_{x_{i+\frac{1}{2}}}^{x_{i+\frac{3}{2}}} + \frac{1}{3}C_2(y,z)x^3|_{x_{i+\frac{1}{2}}}^{x_{i+\frac{3}{2}}} + \frac{1}{4}C_3(y,z)x^4|_{x_{i+\frac{1}{2}}}^{x_{i+\frac{3}{2}}} \tag{B.3}$$

$$q_i(y,z) = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(x,y,z)dx$$
$$= C_0(y,z)x|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} + \frac{1}{2}C_1(y,z)x^2|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} + \frac{1}{3}C_2(y,z)x^3|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} + \frac{1}{4}C_3(y,z)x^4|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \tag{B.4}$$

$$q_{i-1}(y,z) = \int_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} q(x,y,z)dx$$
$$= C_0(y,z)x|_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} + \frac{1}{2}C_1(y,z)x^2|_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} + \frac{1}{3}C_2(y,z)x^3|_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} + \frac{1}{4}C_3(y,z)x^4|_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} \tag{B.5}$$

(B.6)

Solving the above equations for the interfacial value $C_0(y,z)$, we get

$$C_0(y,z) = q_{i+1/2}(y,z) = \frac{7}{12}(q_i(y,z) + q_{i+1}(y,z)) - \frac{1}{12}(q_{i-1}(y,z) + q_{i+2}(y,z)) \tag{B.7}$$

The coefficients can also be obtained by Taylor series expansion of variable $q(x,y,z)$ along $x-$axis and integrating the cell analytically to get the discretized cell averaged

$$q_i(y,z) = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(x,y,z)dx \tag{B.8}$$

which yields

$$\begin{aligned} q_{i+1/2}(y,z) &= q_i(y,z) + q_{i+1}(y,z) + \frac{1}{4}q''(8)(\Delta x)^3 \\ &= q_{i-1}(y,z) + q_{i+2}(y,z) + \frac{1}{4}q''(56)(\Delta x)^3 \end{aligned} \tag{B.9}$$

Then use Richardson extrapolation to obtain the above equation (B.7). Finally, integrate over the control volume face $\sigma(y,z)$ to obtain cell face average,

$$\iint_{\sigma} q_{i+1/2}(y,z)dydz = \frac{7}{12}\iint_{\sigma}(q_i(y,z)+q_{i+1}(y,z))dydz - \frac{1}{12}\iint_{\sigma}(q_{i-1}(y,z)+q_{i+2}(y,z))dydz \tag{B.10}$$

or in the notation of equation (3.9):

$$\overline{q}_{i+1/2,j,k} = \frac{7}{12}(Q_{i,j,k} + Q_{i+1,j,k}) - \frac{1}{12}(Q_{i-1,j,k} + Q_{i+2,j,k}) \tag{B.11}$$

Sanderson & Brassington (1998) obtained the above face-averaged quantity using a cumulative sum of the cell-averaged quantity. The current derivation is complete and can be extended to non-uniform grids when the expansion ratio is known. The truncation error of various differencing operators is shown in Figure B.2 in the example that a sinusoidal signal is numerically differentiated Sanderson & Brassington (1998). The error is plotted as a function of the ratio of grid spacing Δ to wavelength λ. The power-law scaling of error with the ratio λ/Δ is closely followed for wavelengths greater than 4Δ with C-grid differencing and for wavelengths greater than 6Δ with A-grid Sanderson & Brassington (1998). Further, the scaling applies with reasonable approximation to the smallest wavelengths resolvable on the grid. The error analysis and numerical examples in Sanderson & Brassington (1998) demonstrate that the fourth-order interpolation used in the DieCAST model is fourth-order accurate on both A-and C-grids.

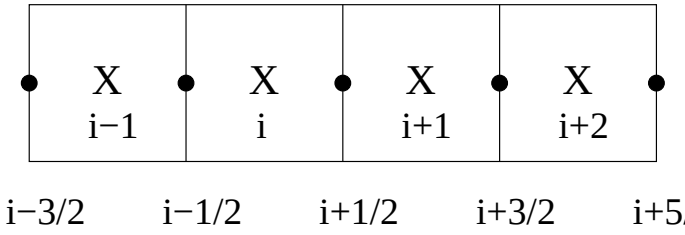


Figure B.1: Schematic of the cells and the index for the fourth-order interpolation.

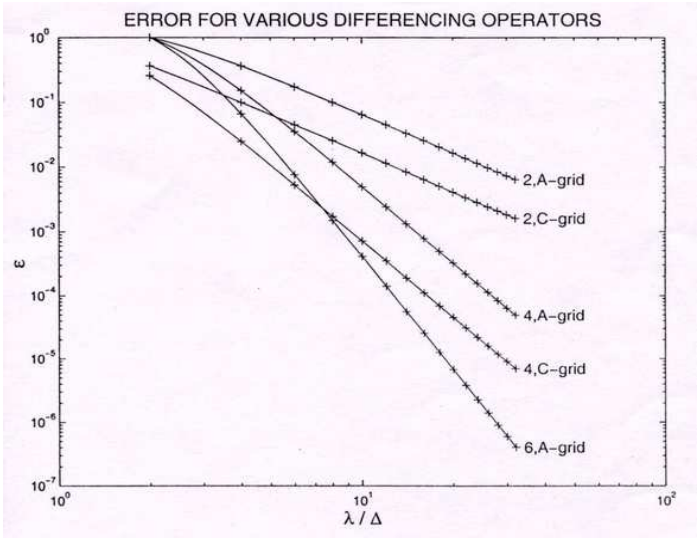


Figure B.2: Error ε for differentiating s sinusoidal function using the A-grid (collocated) and C-grid (staggered) differencing operators. The error is plotted as a function of the ratio of grid spacing Δ to wavelength λ. The label 2,A-grid refers to the second-order operator on a collocated grid. The label 2,C-grid refers to the second-order operator on a staggered grid. Similarly, label 4,A-grid refers to the fourth-order operator on a collocated grid and label 4,C-grid refers to the fourth-order operator on a staggered grid. The sixth-order differencing operator on the A grid is labeled 6,A-grid. [Figure 1 of Sanderson & Brassington (1998)].

Appendix C

Vertical mixing scheme: Pacanowski & Philander (1981)

Vertical viscosity and diffusivity A_v and K_v are given by the formulae:

$$A_v = 0.02 + min(10, 100R^2) + R|W|dz/Re \tag{C.1}$$

$$K_v = 0.02 + Rmin(10, 100R^2) + R|W|dz/ReR \qquad = \frac{1}{1+5Ri} \tag{C.2}$$

$|W|$ is the vertical velocity magnitude, dz is the vertical grid interval, Ri is the gradient Richardson number, and Re is the specified vertical cell Reynolds number. This is simply a non-dimensional ratio of advection to turbulent diffusion for variations having the scale of order one model grid interval.

The first term in Eqns (C.1) and (C.2) represent laminar diffusivity. The second one is a standard Pacanowski and Philander mixing parameterization. The third js a generally small mixed physical/numerical term, assigned a value of 10, which avoids cell Reynolds number overshoots by vertical advection, such as produced by internal waves. It allows extremely small vertical mixing necessary to maintain the cold intermediate layer throughout the model year. The physical processes represented by terms two and three are the vertical Reynolds stress terms representing the nonlinear interaction of subgrid-scale flow components with resolved scales. These diffusion terms are conventional ad hoc representations of the subgrid-scale effects which have limited basis outside of boundary layers.

Appendix D

DieCAST model programs and subroutines

D.1 Listings of the DieCAST model and its preprocessor and postprocessor codes

D.1.1 PREP/prep.f

Program **prep.f** is the preprocessor for DieCAST. It resides in subdirectory PREP. It sets all scalar and array parameters and writes them to output files that are read by DieCAST. The user may write his own program(s) or subroutines to create surface forcing, open lateral boundary and depth data files. The subroutines in **prep.f**:

BLOCK DATA NISHAL initializes all scalar control parameters plus the one-dimensional array 'IE' used by the elliptic rigid lid pressure solver. It also has comments describing the variables to be initialized.

PROGRAM PREP is the main program of **prep.f**. It opens input and output data files and controls the initialization process.

SUBROUTINE SETZW generates the vertical grid (z-levels) used by DieCAST.

SUBROUTINE PRE is the preprocessor for the elliptic rigid lid EVP pressure solver. It generates influence coefficient and inverse matrix data for the EVP solver.

SUBROUTINE INITFS initializes all controlling arrays and also some derived scalar control parameters.

SUBROUTINE BOUNDS is used to derive thermal winds on the logigically rectangular model domain boundaries from Levitus climatology and uses this to derive normal inflow velocities.

SUBROUTINE POISOL is used to fill in T and S values at points not included in Levitus climatology. The model grid might not have the same resolution as Levitus climatology. Also, there is a mismatch between Levitus climatology and bottom topography data resolutions (great savings would

be achieved if someone would fix this). POISOL can be used to fill in the missing data.

D.1.2 THE MODEL

DieCAST reads output files from **prep.f** which specify initial and boundary conditions and other model parameters. Although it is often possible to specify all controlling data in **prep.f**, it is sometimes desirable to modify DieCAST slightly in order to implement special boundary treatments for certain problems.

Subroutines in the Model

MAIN PROGRAM controls the flow of the calculation (and the calculation of the flow). Input and output files are opened; model initialization, time marching, and graphics output subroutines are called; and (optionally) restart data is saved at specified intervals. Restarts are seamless.

SUBROUTINE FS is called by the MAIN PROGRAM to march one time step each time it is called.

The following four subroutines are called optionally by SUBROUTINE FS to allow the user to determine the level of physics desired.

SUBROUTINE DIFUSE adds turbulence contributions to the vertical momentum, heat, and salinity diffusivities. The diffusivities are composed of this optional part plus a user-specified background value plus a part determined by the user-specified vertical cell Reynolds number.

SUBROUTINE WIND adds wind forcing to the top layer.

SUBROUTINE HEAT adds surface heat flux to the top layer, normally from air-sea exchanges.

SUBROUTINE SUN adds radiative heating effects distributed vertically according to a user specified vertical profile array.

SUBROUTINE FSPLTS is called by the MAIN PROGRAM in order to save data for the standard output fields at user specified intervals. Data is saved for the three basic kinds of cross-sections (x-y, x-z and y-z), each at specified intervals in the respective third direction. These include barotropic streamfunction (calculated diagnostically only for graphics output), vorticity, pressure, velocity, temperature, salinity and others.

SUBROUTINE COMXY is (optionally) called by the MAIN PROGRAM to generate output used in animations.

The following three subroutines are called by SUBROUTINE FSPLTS to save data for plots.

SUBROUTINE XYPLOT saves horizontal plots data.

SUBROUTINE XZPLOT saves vertical-longitudinal cross-section plots.

SUBROUTINE YZPLOT saves vertical-latitudinal cross-section plots.

SUBROUTINE RANGE determines the maximum and minimum of the input field data.

SUBROUTINE REP solves the Poisson equation for the rigid lid surface pressure using the EVP method.

SUBROUTINE INITFS is called by the MAIN PROGRAM to read input data files from the preprocessor program PREP.F and do other initialization and optional restart functions.

Appendix E

ALPHABETICAL LISTING OF VARIABLES AND THEIR MEANINGS

This list includes only variables in common. Local variables are used mainly for scratch or are secondary in nature. Asterisks precede user- specified quantities. All others are derived by the preprocessor or by the DieCAST model. Many variables are used only for diagnostics.
A: total area of wet points in each layer
AB: left (west) coefficient in Poisson pressure operator
AC: central coefficient in Poisson pressure operator
AB: bottom (south) coefficient in Poisson pressure operator
AR: right (east) coefficient in Poisson pressure operator
AT: top (north) coefficient in Poisson pressure operator
AROUT: area of boundary region to be adjusted to get zero net inflow
AV: current number of time levels in long term average
AVGXYT: horizontal mean temperature and salinity data
*B: coefficient of thermal expansion for water (in eq'n of state)
CS,CSV: spherical coordinate latitude cosine factors
*DAODT: number of time steps per model day
DAYS: present time in days
*DE0: background horizontal diffusivity
DHX,DHY: x-, y-directed turbulent diffusivity
*DM0: background horizontal viscosity
DMX,DMY: x-, y-directed turbulent viscosity
DMZ0: background vertical viscosity
*DRAG: bottom drag coefficient
DT: time step size
DUM0,DUM1,DUM2: scratch arrays
DX,DY,DXV,DYV: lateral grid increments across cell-centers and grid lines
EV: vertical turbulent viscosity array
EVAP: integrated E-P volume flux

F: Coriolis parameter
*FLTW: coefficient in fltw time marching method (Asselin filter)
FNEW,FOLD: time interpolation factors between months NEW,NLD
*G: gravity
GB: gravity times coefficient of thermal expansion
H: scratch array used by surface pressure Poisson solver
HV: vertical turbulent diffusivity array
*IE: array of evp block boundaries used in Poisson pressure solver
IN: mask array for scalar quantities
INFX,INFY: boundary mask arrays for biharmonic filter
*IPLOT: time step between main graphics data outputs
IT0: first time step of present run (=0 or restart time step number)
ITF: time step number
ITFDAY: number of time steps per day
IU0: 2-d mask array for staggered x-component advection velocity (applies to top layer only)
IU: 3-d mask array for staggered x-component advection velocity
IV0: 2-d mask array for staggered y-component advection velocity (applies to top layer only)
IV: 3-d mask array for staggered y-component advection velocity
IW: 3-d mask array for staggered z-component advection velocity
KB: 2-d array, number of levels containing water (logical depth)
*KTRM: thermocline level (used only for animation graphics)
*LHEAT: flag for surface heating (air-sea exchange)
*LMOVI: flag to save data for animation
*LOPEN: flag for open boundary conditions
*LRSTRT: flag to initialize from restart file
*LSAVE: flag to save restart file
*LSOL: flag for radiative heating (vertical profile)
*LTURB: flag for call turbulence submodel subroutine
*LWIND: flag to include wind forcing
*M1,M2,M3,M4,M5,M6,M7,M8,M5M: miscellaneous output frequency controls
*MVI: number of time steps between movie frame storage
*MXIT: last time step of present run (time step number is saved and not reset to zero in restarts)
*MXPLOT: maximum number of time steps between main graphics data outputs
N360: present time in days after start of present model year
NEW,NLD: present, previous month in annual cycle
NSOMBO: number of ensemble-averaging months to present
OCS,OCSV: spherical coordinate latitude inverse cosine factors
ODX,ODY,ODV,ODXV,ODT: inverse horizontal grid increments and time step
ODZ: inverse layer thickness array (distance between w-levels)

ODZW: inverse distance between layer pressure levels
OFLTW: unity minus FLTW
OLDE, OLDQ: previous months E-P and surface Q
ORZMX: inverse of specified vertical cell Reynolds number
P0: pressure against rigid lid or, hydrostatically, surface height
P: pressure anamoly (deviation from base hydrostatic)
PBAR: time mean surface pressure
PRN: Prandtl number (DM0/DE0)
PVAR: rms deviation of surface pressure from its time mean
PX,PY: scratch arrays for horizontal pressure gradient
QAVG: multi-year ensemble monthly time mean surface heating
QPROF: vertical radiative heating profile array
QSUM: mean surface layer heating (watts/m-m)
RHO: potential density deviation from minimum potential density
RINV1: influence coefficient array for elliptic pressure solver
RINV: inverse array for elliptic pressure solver
RMSV: mean squared velocity anomaly at 700m depth
RNN: array used to blend 2nd- and 4th- order p.g. near boundaries
RZMX: specified vertical cell Reynolds number
S1,S2,SLF: old, filtered (central), and leapfrog salinity fields
S: source term for rigid lid pressure Poisson equation
SBAR: time mean salinity
SCLI: annual mean climatological salinity
SCR: 3-d multi-purpose scratch array
SSURF,TSURF: monthly surface climatology salinity and temperature
SSURFM, TSURFM: horizontal mean monthly surface climatology
SUMIN: number of "wet points" in each model layer
SZ: z-component of salinity flux (advection only)
TANPHI: tangent of latitude
T1,T2,TLF: old, filtered (central), and leapfrog T fields
TAUDT: surface layer restoring (nudging) factor
*TAU: surface restoring time in days
*TAUN: lateral boundary restoring time in days
TAVG: horizontal mean temperature
TBAR: time mean temperature
TCLI: annual mean climatological temperature
TLZ: maximum depth of modeled domain (deeper water depths are set to TLZ)
TNUDGE: summed surface layer temperature nudging during present monnth
TSP: longitudinally averaged monthly temperature climatology
TW: running time averages used in vertical heat flux diagnostics
TX: x-component of total (advection + diffusion) heat flux
TY: y-component of total (advection + diffusion) heat flux
TZ: z-component of heat flux (advection only)

U1,U2,ULF: old, filtered (central), and leapfrog x-velocity fields
U: staggered "c" grid x-component non-divergent advection velocity
UCLI: time mean longitudinal velocity at 700m depth
UX: x-component of total (advection + diffusion) u-momentum flux
UY: y-component of total (advection + diffusion) u-momentum flux
UZ: z-component of u-momentum flux (advection only)
V1,V2,VLF: old, filtered (central), and leapfrog y-velocity fields
V: staggered "c" grid y-component non-divergent advection velocity
VCLI: time mean latitudinal velocity at 700m depth
VX: x-component of total (advection + diffusion) v-momentum flux
VY: y-component of total (advection + diffusion) v-momentum flux
VZ: z-component of v-momentum flux (advection only)
W: vertical velocity
WATTS: conversion factor from (deg C per model time step) to $watts/m^2$
WAVG: multi-year ensemble monthly time mean precipitation-evaporation
X: scratch array used by surface pressure Poisson solver
XBAR: time mean streamfunction for vertically averaged flow
Y: latitude cell centers
YV: latitude grid lines
Z: stretched vertical grid array (includes both pressure levels and layer interface levels in increasing depth sequence)

Bibliography

Asselin, R. 1972. Frequency filter for time integrations. *Monthly Weather Review*, **100**, 487–490.

Cushman-Roisin, B., Korotenko, K. A., Galos, C. E., & Dietrich, D. E. 2004. Mesoscale simulations of the Adriatic Sea variability. submitted to J. Geophysical Research.

Dietrich, D. E. 1997. Application of a modified 'a' grid ocean model having reduced numerical dispersion to the gulf of Mexico circulation. *Dyn. Atmos. Oceans*, **27**, 201–217.

Dietrich, D. E., & Lin, C. A. 2002. Effects of hydrostatic approximation and resolution on the simulation of convective adjustment. *Tellus*, **54(A)**, 34–43.

Dietrich, D. E., Marietta, M. G., & Roache, P. J. 1987. An ocean modeling system with turbulent boundary layers and topography: Part 1. Numerical studies of small island wakes in the ocean. *Int. J. Numer. Methods Fluids*, **7**, 833–855.

Dietrich, D. E., Bowman, M. J., Lin, C. A., & Mestas-Nunez, A. 1996. Numerical studies of small island wakes in the ocean. *Geophys. Astrophys. Fluid Dyn.*, **83**, 195–231.

Dietrich, D. E., Haney, R. L., Fernandez, V., Posey, S., & Tintore, J. 2003. Model-determined surface heating and freshwater sources using a precise, non-damping nudging approach. *J. Mar. Systems (accepted)*.

Dietrich, D. E., Mehra, A., Haney, R. L., Bowman, M. J., & Tseng, Y. H. 2004. Dissipation effects in modeling the North Atlantic ocean. *Geophys. Res. Letters*, **31**, L05302.

Ezer, T. 2004. Entrainment, diapycnal mixing and transport in three-dimensional bottom gravity current simulations using the Mellor-Yamada turbulence scheme. *Ocean Modelling*, in press.

Ezer, T., & Mellor, G.L. 2004. A generalized coordinate ocean model and a comparison of the bottom boundary layer dynamics in terrain-following and in z-level grid. *Ocean Modelling*, **6**, 379–403.

Girton, J. B., & Sanford, T. B. 2003. Descent and modification of the overflow plume in the Denmark Strait. *J. Phys. Oceanogr.*, **33**, 1351–1364.

Haney, R. L. 1971. Surface thermal boundary condition for ocean circulation modes. *J. Phys. Oceanogr.*, **1**, 241–248.

Pacanowski, R. C., & Philander, S. G. H. 1981. Parameterization of vertical mixing in numerical models of tropical ocean. *J. Phys. Oceanogr.*, **11**, 1443–1451.

Roache, P. J. 1995. *Elliptic marching methods and domain decomposition.* CRC Press.

Roache, P. J., & Dietrich, D. E. 1988. Evaluation of the filtered Leapfrog-Trapezoidal time integration method. *Num. Heat Transfer*, **14**, 149–164.

Sanderson, B. G., & Brassington, G. 1998. Accuracy in the context of a control-volume model. *Atmos.-Ocean*, **36**, 355–384.

Tseng, Y. H., & Ferziger, J. H. 2003. A ghost-cell immersed boundary method for flow in complex geometry. *J. Compu. Phys.*, **192**, 593–623.

Tseng, Y. H., & Ferziger, J. H. 2004. Large-eddy simulation of turbulent wavy boundary flow - illustration of vortex dynamics. *J. Turbulence*, (in press).

Tseng, Y. H., Dietrich, D. E., & Ferziger, J. H. 2004. Regional circulation of the Monterey Bay region- The effects of Monterey Canyon. *Submitted to J. Geophy. Res.-Ocean.*