U. S. DEPARTMENT OF COMMERCE
NATIONAL OCEAN AND ATMOSPHERIC ADMINISTRATION
NATIONAL WEATHER SERVICE
NATIONAL CENTERS FOR ENVIRONMENTAL PREDICTION

TECHNICAL NOTE

COASTAL OCEAN FORECASTING SYSTEM (COFS) SYSTEM DESCRIPTION AND USER GUIDES

LECH LOBOCKI
OCEAN MODELING BRANCH

JULY 1996

---

*OMB CONTRIBUTION NO. 127

OPC CONTRIBUTIONS

No. 1. Burroughs, L. D., 1987: Development of Forecast Guidance for Santa Ana Conditions. National Weather Digest, Vol. 12 No. 1, 7pp.

No. 2. Richardson, W. S., D. J. Schwab, Y. Y. Chao, and D. M. Wright, 1986: Lake Erie Wave Height Forecasts Generated by Empirical and Dynamical Methods -- Comparison and Verification. Technical Note, 23pp.

No. 3. Auer, S. J., 1986: Determination of Errors in LFM Forecasts Surface Lows Over the Northwest Atlantic Ocean. Technical Note/NMC Office Note No. 313, 17pp.

No. 4. Rao, D. B., S. D. Steenrod, and B. V. Sanchez, 1987: A Method of Calculating the Total Flow from A Given Sea Surface Topography. NASA Technical Memorandum 87799., 19pp.

No. 5. Feit, D. M., 1986: Compendium of Marine Meteorological and Oceanographic Products of the Ocean Products Center. NOAA Technical Memorandum NWS NMC 68, 93pp.

No. 6. Auer, S. J., 1986: A Comparison of the LFM, Spectral, and ECMWF Numerical Model Forecasts of Deepening Oceanic Cyclones During One Cool Season. Technical Note/NMC Office Note No. 312, 20pp.

No. 7. Burroughs, L. D., 1987: Development of Open Fog Forecasting Regions. Technical Note/NMC Office Note. No. 323., 36pp.

No. 8. Yu, T. W., 1987: A Technique of Deducing Wind Direction from Satellite Measurements of Wind Speed. Monthly Weather Review, 115, 1929-1939.

No. 9. Auer, S. J., 1987: Five-Year Climatological Survey of the Gulf Stream System and Its Associated Rings. Journal of Geophysical Research, 92, 11,709-11,726.

No. 10. Chao, Y. Y., 1987: Forecasting Wave Conditions Affected by Currents and Bottom Topography. Technical Note, 11pp.

No. 11. Esteva, D. C., 1987: The Editing and Averaging of Altimeter Wave and Wind Data. Technical Note, 4pp.

No. 12. Feit, D. M., 1987: Forecasting Superstructure Icing for Alaskan Waters. National Weather Digest, 12, 5-10.

No. 13. Sanchez, B. V., D. B. Rao, and S. D. Steenrod, 1987: Tidal Estimation in the Atlantic and Indian Oceans. Marine Geodesy, 10, 309-350.

No. 14. Gemmill, W. H., T. W. Yu, and D. M. Feit 1988: Performance of Techniques Used to Derive Ocean Surface Winds. Technical Note/NMC Office Note No. 330, 34pp.

No. 15. Gemmill, W. H., T. W. Yu, and D. M. Feit 1987: Performance Statistics of Techniques Used to Determine Ocean Surface Winds. Conference Preprint, Workshop Proceedings AES/CMOS 2nd Workshop of Operational Meteorology. Halifax, Nova Scotia., 234-243.

No. 16. Yu, T. W., 1988: A Method for Determining Equivalent Depths of the Atmospheric Boundary Layer Over the Oceans. Journal of Geophysical Research. 93, 3655-3661.

No. 17. Yu, T. W., 1987: Analysis of the Atmospheric Mixed Layer Heights Over the Oceans. Conference Preprint, Workshop Proceedings AES/CMOS 2nd Workshop of Operational Meteorology, Halifax, Nova Scotia, 2, 425-432.

No. 18. Feit, D. M., 1987: An Operational Forecast System for Superstructure Icing. Proceedings Fourth Conference Meteorology and Oceanography of the Coastal Zone. 4pp.

No. 19. Esteva, D. C., 1988: Evaluation of Priliminary Experiments Assimilating Seasat Significant Wave Height into a Spectral Wave Model. Journal of Geophysical Research. 93, 14,099-14,105.

No. 20. Chao, Y. Y., 1988: Evaluation of Wave Forecast for the Gulf of Mexico. Proceedings Fourth Conference Meteorology and Oceanography of the Coastal Zone, 42-49.

Coastal Ocean Forecasting System (COFS)

Design and Operation of the Model Execution Suite

Lech Lobocki
Environmental Modeling Center
Ocean Modeling Branch

Release 3.1
Washington, May 29, 1995

Abstract

This note consists of five documents, assembled to provide comprehensive documentation for the Coastal Ocean Forecasting System (COFS), running at the National Centers for Environmental Prediction:

The *COFS Design and Operation of the Model Execution Suite*, by L. Lobocki, documents the data management and model execution system that provides the model with continuous atmospheric forcing.

The *COFS input archive*, by L. Lobocki, describes the structure and operation of the input archive, which contains selected forecast fields obtained with the atmospheric forecast model.

A Description of a Three-Dimensional Coastal Ocean Circulation Model, by A.F. Blumberg and G.L. Mellor, presents the foundations of the model.

The *User's Guide for a Three-Dimensional, Primitive Equation, Numerical Ocean Model*, by G.L. Mellor, contains a model description along with documentation of major code modules.

A User's Manual for the Princeton Numerical Ocean Model by W.P. O'Connor contains a detailed documentation of the model code.

1. Introduction

This document describes current status of the Coastal Ocean Forecast System (COFS) running semi-operationally at the Environmental Modeling Center, NCEP/NWS/NOAA, during its implementation phase. The core of the system is a three-dimensional, primitive equation model of ocean dynamics, commonly known as the Princeton Ocean Model (POM). The current model configuration uses atmospheric forcing as a principal mechanism of evolution; at present, there is no real-time data assimilation, and the forcing on open boundaries is based on climatology. At the time of writing, the data assimilation is under development.

During the initial phase of the project, the model was run in either a hindcast or forecast mode, using climatological fields as initial conditions, and atmospheric forcing produced by the 80-km, 38-level version of regional/mesoscale 'Eta' model, which has been used operationally at the NCEP between 1993 and 1995. This situation changed in 1995, when a new, 29-km, 50-level mesoscale version of the Eta model became operational. Resulting changes in related data structures, together with changes in NCEP's operational suite and with certain extensions of the model and its output handling, required a more complex of data handling scheme. This led to a next generation data processing system centered around the model which is described in this document. For a description of the model itself, see Blumberg and Mellor (1987), Mellor (1996), and O'Connor (1991). These documents are retained here as appendices, along with a description of the input archive.

2. Operational requirements and system design

The principal purpose of the data management system described here is to provide a continuous stream of atmospheric data to the model in a suitable format, and to control execution of the model in both the hindcast and operational modes. Next, post-processing of the model output is required in order to reduce the amount of data, and to arrange data into convenient structures, coded in machine-portable, standardized formats.

Information on atmospheric forcing data comes from the separatedly maintained COFS input archive, containing subsets of Eta forecasts created immediately after individual runs by the archiving software. These data form daily files, containing collections of selected prognostic fields, coded in GRIB format (Stackpole, 1994), from both 00Z and 12Z Eta model runs, arranged into monthly subdirectories. The basic role of the data processing system is to locate required data, decode and transform them onto the COFS grid, and prepare control information as required by the model. Next, the system has to execute the model code, and finally transform selected results into desired formats, storing them in the model output archives. An additional requirement is to gather diagnostic information, generated by individual system components, and to provide it in a form suitable for both supervising the system and error tracing.

Experience gained during the first phase of COFS operations showed that a large percentage of model crashes was due to occasional gaps in the input data stream. Therefore, one of the design goals was to address this problem through a more systematic approach, and to provide for fault-tolerant model execution. Next, it was desired to provide for executing the system in both the forecast and hindcast modes, and finally, to

make the system capable of recognizing and adjusting to different grids used by the Eta model.

To address these requirements, a multicomponent system was designed and created. The system consists of the main control program, written in C language, and a set of various programs, written mainly in FORTRAN, which accomplish individual smaller tasks. The main computational platform is a Cray C-90 series computer running under UNICOS 7.x (or later) operating system.

Due to computational demand, the system is primarily designed to be run in a batch mode. However, the data availability analysis together with interpolation constitute a shorter task which may be executed separatedly, within an interactive session. For this reason, model execution is controlled by batch job script, generated and submitted automatically for batch execution by the system upon completion of the data preparation stage. Execution of the control program may be either initiated manually or chained to the production runs so that it starts automatically after input archives are updated. To facilitate maintenance, the system incorporates a multi-level tracing system and a mail notification feature.

The system supports decoding and coding the data in GRIB format, which is applied to all 2-D horizontal fields for both the input and output. Several files containing constant input information, such as gridpoint locations, input climatologies, etc., are stored in the machine native binary format (they are essentially not handled by the system). Some of the smaller output files are in ASCII; certain temporary files intended for internal use also remain in machine native binary format. The model output processing accomodates seven different categories, characterized by different volume, frequency and destination. These are intended to satisfy both operational requirements as well as extended diagnostics demands. A suite of various batch-mode and interactive graphic programs exists (with a number of suitable decoders/converters), running on platforms including Cray, UNIX workstations, and PC.

To deal with missing data, a hierarchical data availability analysis and substitution scheme was designed, which allows for automatic substitution of missing data which would be 'normally' used with data originating from other forecast cycles covering fully (or partially) the required period. With a 36-hour extent of the Eta-MESO forecasts, executed twice per day (00Z and 12Z) and a 24-hr COFS forecast horizon, the system can survive gaps up to two cycles wide. Thought the implementation of the selection scheme using preference tables, particular choices can be reprogrammed easily for various possible model setups and situations.

System execution consists of four phases, which are discussed in detail in section 4.1:

- request analysis and preliminary decision on the run extent;
- data availability analysis
- interpolation
- batch script generation for execution of the model and its submission.

Phases 2 and 3 are repeated at every forecast time; phase 4 also includes post-processing the model output which will be discussed separatedly.

3. Operational (forecast) and hindcast runs

The system can run in either the hindcast, or in the operational (production) mode. Since the present version of the COFS does not incorporate data assimilation, both the hindcast and forecast runs normally use the same data (differences may arise when some data are missing). Each mode differs in data availability and the allowed extent of the model run. Operational runs are generally restricted to 24 hr. (in certain situations they may be substituted by two 12-hr. runs) while hindcast runs may cover longer periods, usually limited by CPU time restrictions and input data availability. Operational runs are generally intended for automatic launching, which is initiated by the production suite, after completion of the atmospheric model execution (the Eta-MESO). Hindcast runs are initiated manually and then may be automatically chained to facilitate longer runs. In the hindcast mode, the system is allowed to use data from 'future' forecast cycles to fill gaps in the stream when 'present' data are missing, whereas in the operation mode, data from future runs are not available at execution time. In addition, the control program provides for different output processing for operational runs and for hindcast runs.

4. Overview of software components

4.1 The control program

4.1.1 Request analysis

Control information, passed to the main control module, consists of four specified elements:

- Intended run extent[1] (number of days)
- Execution mode (hindcast/operational)
- Starting date (initial conditions)
- Current date.

Additionally, certain information such as file location and naming is embedded in the control module code in a section for separate settings.

Intended run length and the execution mode are passed as command-line parameters (see section 7.1). To determine the starting time of the run, the system utilizes a restart file name, which contains a date/time stamp. Current date/time is retrieved from the system clock.

The request analysis scheme, coded as a separate routine named *RequestAnalysis*, performs the following steps:

[1] The actual run extent is determined automatically later on the basis of data availability.

- checks the availability of individual directories.
- parses the command line
- locates the current restart file and decodes the time/date stamp from its name.
- rejects or ignores runs when the the restart file is up to date.
- performs preliminary determination of the run extent.

For detailed description of these steps, see section 7.2.

4.1.2 Data availability analysis

Data availability analysis (DAA) is done within a time loop, stepping through all the run extent as specified at the request analysis phase. It is performed alternately with interpolation, until either reaching the intended end of the run period, or finding a gap in the data which cannot be filled by using data from other Eta forecast cycles. In the latter case, either the run extent is shortened, or the execution is aborted.

Each of the DAA steps involves scanning the archive in search of required files, and checks for the presence of needed fields in these files. The archive files are daily collections of relevant GRIB messages, created everyday during the 00Z cycle and then amended during the 12Z cycle postprocessing. The DAA thus has to determine which files should be scanned, which messages are to be searched for, and what to do if the data are not found. The present version of the control system utilizes a priority table-based search (Figure 4-1) which ensures a continuous information stream when the gaps in the archive are **not wider than two consecutive cycles**. The procedure involves three scanning steps. The data availability analysis is discussed in more detail in chapter 7.1.

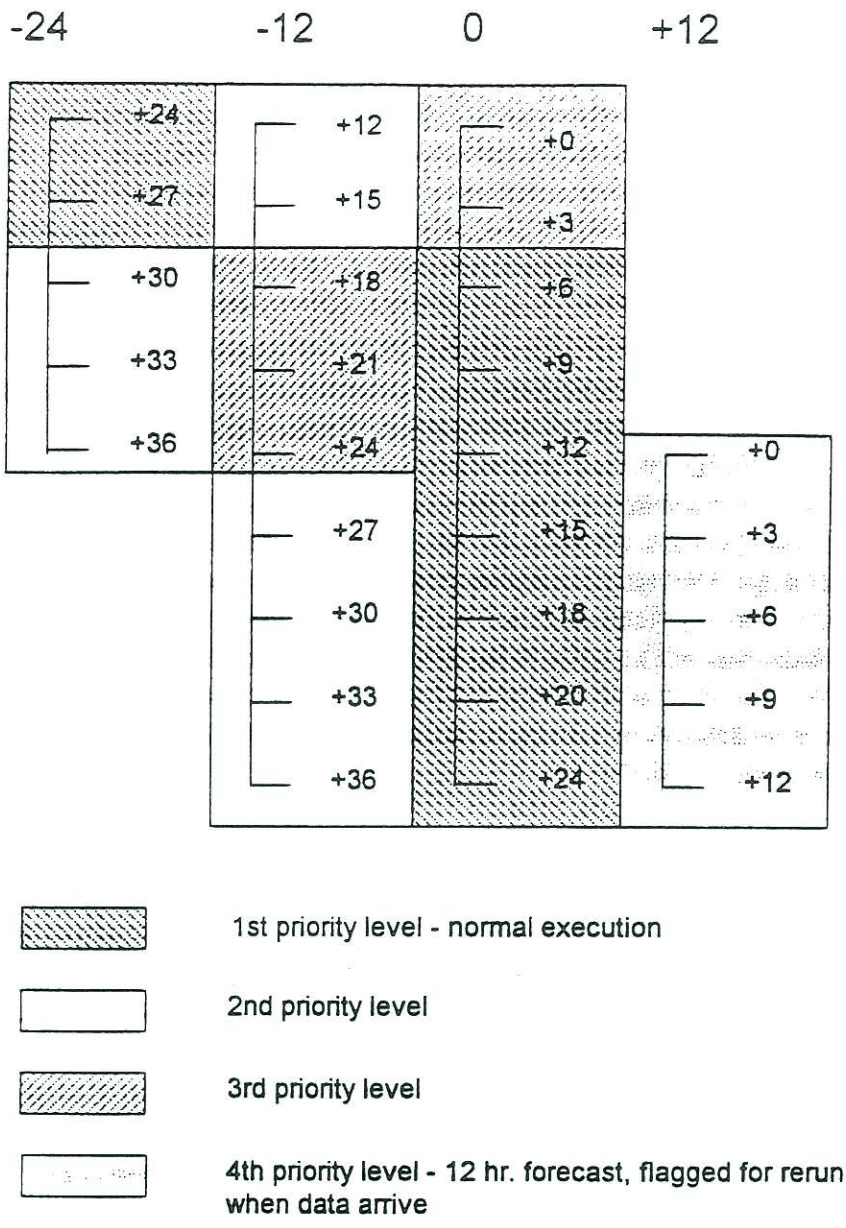


Figure 4-1. Data availability diagram and priority scheme for input data selection.

4.1.3 Decoding & interpolation phase

During this phase, data selected by the data availability analysis scheme are extracted and decoded from the archival GRIB files, and then interpolated from their original grid to the COFS domain. Subsequent meteorological fields are then written to the output file, which constitutes the main atmospheric input field for the model.

In this document, the interpolation code itself, is not documented, as a separate utility; rather, we restrict discussion to a description on how this utility is controlled by the higher-level control program. However, it is necessary to mention that the pre-1995 version of the interpolation code has been extended by: (1) incorporation of GRIB

decoding, and (2) automated Eta grid recognition and generation. A brief discussion of these features along with a description of the interpolation code and its usage may be found in section 4.3. In addition, the interpolation code calculates area-averaged values of surface fluxes, which are stored in a direct access file. The model code, as used in versions prior to 3.0, utilized these values for flux corrections; a separate utility was used to construct longer time series. This became impractical when the data migration facility access deteriorated, and, as a result, the code was equipped with an ability to maintain a database-style file with record append and replace functions available. This file is ASCII-coded and provides a quick, convenient way of monitoring the flux climatology.

Starting from version 3.0, the COFS/POM incorporated changes in salinity fields due to evaporation/precipitation. This required precipitation rates retrieved from the Eta model output. Unfortunately, the latter contains accumulated precipitation rather than precipitation rates, so that precipitation rates must be retrieved from subsequent accumulation records. This increases the complexity of the data availability analysis scheme, because (1) the scheme needs to consider and choose two data files, not just one, (2) various versions of the Eta model employ different precipitation accumulation periods, (3) the EDAS analysis files, used as forecast data for +00 hr, differ from the forecast results in that they do not contain valid accumulated precipitation information. In addition, handling of precipitation in the model code differs from that of the instanteneous fields since the precipitation rates which are obtained represent time integrals over a given period rather than rates at given moments.

4.1.4 Model execution and postprocessing

Upon completion of the data availability analysis and the decoding and interpolation, the control program generates and initates a batch job *cfs.qsub* which handles input/output file assignment, model code execution, and postprocessing and disposition of various forms of the model output. Also, this includes mail notification and job chaining for hindcast runs.

The model postprocessing involves several groups of data which are directed in various ways from the model (Table 4-1). A first group of data involves fixed information, such as grid setting, model version ID, etc. The main prognostic fields are outputted as 3-D or 2-D arrays, depending on their nature, both as instantaneous and as average values over a specified time, typically 24 hrs. These data are voluminous. Second group of data contains extreme values of selected fields such as sea surface elevations. This group has a low volume due to the limited number of fields included. Next, due to semidiurnal variablility of the tides, instantaneous fields must be outputted with a higher frequency, say 1-3 hrs. As space is limited, this form of output is restricted to a few 2-D arrays. Further, instantaneous values of surface temperature and sea levels at a limited number of selected gridpoints are written every hour to enable closer verification against measurements. The last group of output files is a set of vertical profiles of multiple variables that can be set at desired locations for diagnostic purposes. Since this channel is expected to operate at a higher frequency than the main 3-D output and to include more variables, it must be linked directly to the model rather than from 3-D output files. In contrast, vertical cross-sections are currently constructed from the 3-D files, so they have low outputting frequency and contain a limited set of variables.

cathegory	frequency	volume	quantities	purpose
2D/3D instantaneous fields - main prognostic variables, TKE, vertically integrated speed, surface elevations	24 hrs	high 3.6 M/day	z_{sfc}, u_B, v_B, u, v, T, S, ω, $q^2/2$, ℓ	routine product, model evaluation
2D/3D average fields - main prognostic variables, vertically integrated speed, surface elevations	24 hrs	high 2.2 M/day	z_{sfc}, u_B, v_B, u, v, T, S	observing long time trends, comparisons
daily max./min. sea elevations (2D)	24 hrs	low 50 K/day	z_{sfc}	routine product
instantaneous surface elevations and currents (1m depth)	1-3 hr	high or moderate 0.7-1.9M/day	z_{sfc}, u, v	routine product
Cross-sections (instantaneous)	24 hrs	moderate (1 M/day)	u, v, T, S	model evaluation
Profiles (selected sites only)	1Δt-6 hrs	moderate	z_{sfc}, u, v, T, S, ω, $q^2/2$, ℓ, K_M, K_H, K_q, Ri, ρ, other	model evaluation
Time series at selected sites	1 hr	low	z_{sfc}, T	model evaluation

Table 4-1. Summary of different categories of the COFS output.

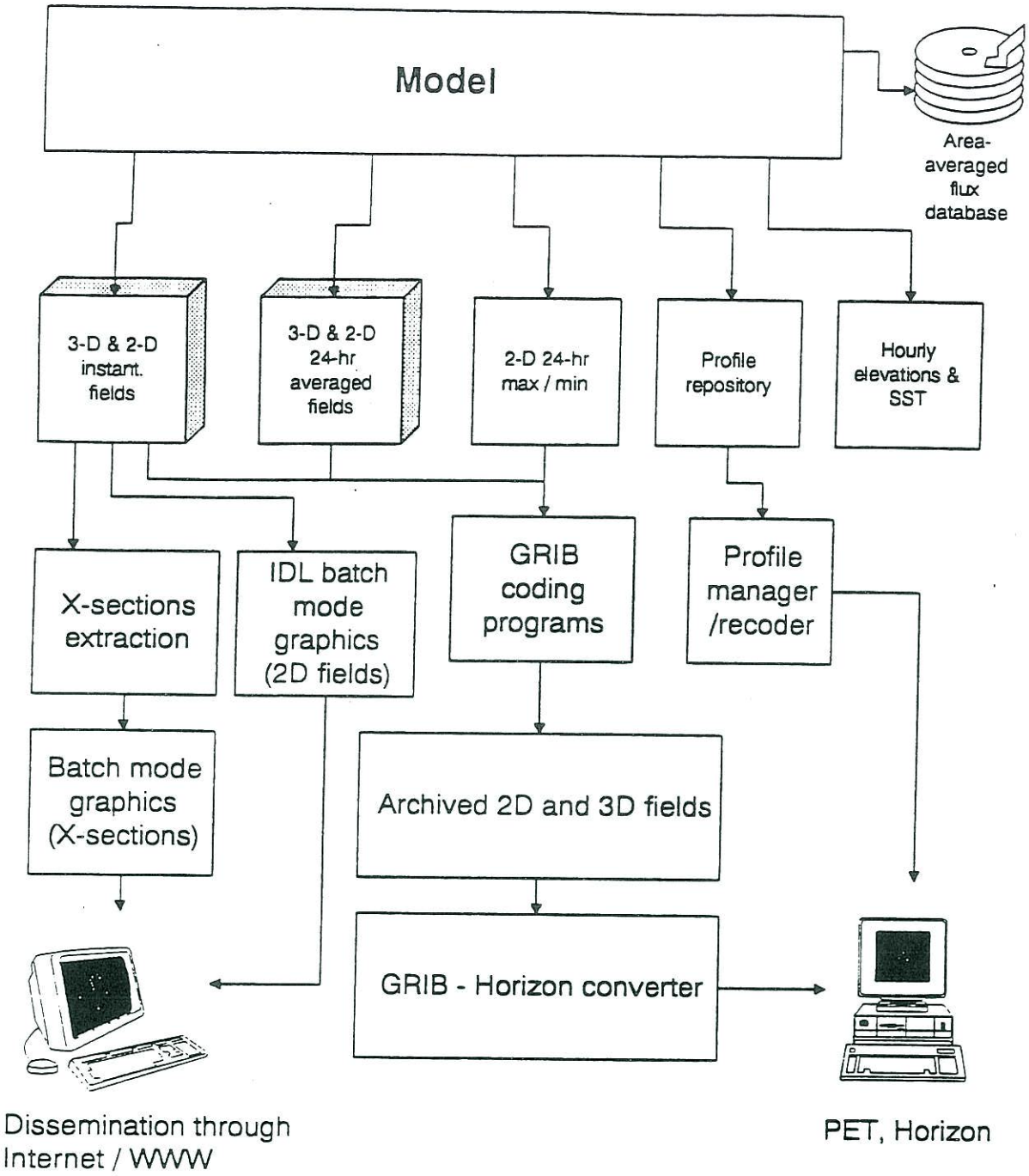


Figure 2. A flowchart of COFS post-processing.

4.2 GRIB manipulation utilities

4.2.1 GRIB indexing program

Purpose: scan a GRIB conglomerate file and create an index file, containing keywords (GRIB PDS) and indexes of individual GRIB messages within the file.

Author: M. Iredell

Usage: `grbindex.x` *GRIB-file-name GRIB-index-file-name*

Returns: 0 when succesful, non-zero otherwise

Default location - executables: /ombptmp/cfspom/tools/grbindex.x

Source code: files contained in /ombptmp/cfspom/tools/windex1.f

Source language: Fortran 77

Machine: Cray Y-MP series

4.2.2 GRIB scan utility

Purpose: search keywords in a GRIB index file to find whether messages fitting description given by a list file and given date/time are present.

Author: L. Lobocki

Usage: `inqgrb.x` *GRIB-index-file-name YYMMDDHH forecast-time cutoff*

where *YYMMDDHH* denotes date and nominal time of the forecast cycle, and *cutoff* is the assimilation cutoff time when the assimilation period is included in the forecast period.

Default location - executables: /ombptmp/cfspom/tools/grbindex.x

Returns: 0 when all the listed data were found, a non-zero value otherwise.

Source code: /ombptmp/cfspom/tools/inqgrb.f, /ombptmp/cfspom/tools/getgb.f

Libraries: /nwprod/w3lib

Source language: Fortran 77

Machine: Cray Y-MP series

4.3 Interpolation code

Purpose: decode a sequence of GRIB-coded fields originating from the Eta model, interpolate it to the COFS grid and write them to the output file.

Authors: D. Sheinin, P.Chen, L. Lobocki

Usage: `intp31.x` *GRIB-file-name1 GRIB-index-file-name1*
GRIB-file-name2 GRIB-index-file-name2
GRIB-file-name3 GRIB-index-file-name3
output-file-name YYMMDDHH
starting-hour final-hour time-step cutoff mode

where *YYMMDDHH* denotes date and nominal time of the forecast cycle, and *cutoff* is the assimilation cutoff time when the assimilation period is included in the forecast period.

Default location - executables: /ombptmp/cfspom/intp/intp.x

Returns:
- 0 when successful, or an error code as follows
- 1 - wrong number of parameters
- 2 - required parameters missing or undecipherable
- 3 - failed to open trace file
- 4 - failed to open input files (grib & index)
- 5 - append mode requested, but output file not found, or failed to open output file
- 6 - unrecognized model id/grid id pair was found in the grib index file
- 7 - problems with opening/writing/reading the temporary grid file
- 8 - problems with opening/reading the CFS grid file
- >100 - error code returned by the w3lib or M. Iredell's getgb.f, plus 100
- 9999 - logical error in program structure. If encountered, notify authors.

Source code: /ombptmp/cfspom/CFS3.1/intp/intp31.f , /ombptmp/cfspom/tools/getgb.f

Source language: Fortran 77

Machine: Cray Y-MP series

Discussion:

The main function of this program is to interpolate meteorological data from the original native Eta grid to the COFS grid. The Eta grid is either generated at run time, or read from a temporary file as discussed later. The COFS grid is read from the *inhts.cfs* file. The code uses a land/sea mask (file *smask.3*), and a gridpoint elimination to reduce noise in coastal areas, and has an ability to extrapolate the data to areas not covered by the Eta grid.

The code attempts to find a time sequence of data from the period from *starting-hour* to *final-hour* with *time-step* and the data assimilation cutoff time given by *cutoff*, in the file specified by *GRIB-file-name1*, using the index file specified by *GRIB-index-file-name1*. Since the precipitation data are available in the Eta model results, as accumulated over a prescribed time periods, it is necessary to calculate precipitation rates. Thus, two subsequent input GRIB files are necessary rather than a single one. Since data availability problems may arise, the interpolation code design does not assume that one of these files must be the one currently processed to retrieve instantaneous fields. As a result, there are three input data sets involved for each interpolation run. *GRIB-file-name1* and *GRIB-index-file-name1* specify files from which instantaneous fields will be retrieved. *GRIB-file-name2* and *GRIB-index-file-name2* are files which contain the previous accumulation records, and *GRIB-file-name3* and *GRIB-index-file-name3* describe files containing current accumulation records. The program subtracts the 'previous' accumulations from the 'current' ones and divides it by the time period to obtain the average precipitation rate.

The *mode* parameter, taking values of either 'N' or 'A', specifies whether the output file specified by *output-file-name* should be opened for write operations as new (thus causing any existing file of this name to be overwritten) or whether an existing file should be opened to append all the subsequent output after its last record.

To eliminate problems with a bias in surface heat fluxes predicted by the pre-1995 (early 80 km) version of the Eta model, the interpolation code was substituting Eta sensible and

latent fluxes with values obtained by a method based on Large and Pond (1981, 1982) using 10 m quantities and the SST. Later, a direct calculation of fluxes in the ocean model, using 10 m quantities and prognostic SSTs was introduced, deactivating this part of the code.

The input data may originate from different configurations of the Eta model, which are related to different grids. The present version of the program has encoded definitions of three operational Eta grids, #90 (80 km early), #94 (29 km meso), and #96 (48 km early) as described in Office Note 388. During first execution of the code, geographic coordinates of individual gridpoints are calculated along with a gridpoint-to-gridpoint assignment table, and interpolation weights. Since these calculations are lengthy, results are stored in files *gridNN.bin* and *gridNN.int* (NN stands for the grid number), which are used during all the subsequent executions.

When interpolation is done, the program calculates area-averaged fluxes over the COFS domain and writes them into a direct access file, *flux.log*. This file is created automatically during first execution of interpolation program, and its first record corresponds to the first date/time of the first processed field for the entire run. Any subsequent execution of the interpolation program will calculate a reference position relative to this first date/time, and will write the flux record at this position (if a data record already exists at this position, it will be overwritten). Negative positions which may result from attempts to execute the model starting with dates prior to the date of establishing the flux log are illegal and may lead to unpredictable program behavior, including damage to the file. To minimize the possible consequences of such errors, the *flux.log* file is backed up in every run into the main run directory as *flux.log.YYMMDD*.

4.4 The Princeton Ocean Model

Authors: A. Blumberg, G.L.Mellor

A description of the model and the code may be found in Blumberg and Mellor, 1987; Mellor, 1996; O'Connor, 1991. Here, we restrict discussion to data files and control information.

The model code utilizes the following data files (i.e., COFS version 3.1):

tide6.bdr - tide parameters at boundaries (unit #7)

tbs_tbe.mon - monthly climatological boundary conditions (unit #8)

els_ele_.2yr - mean boundary elevations (unit #9)

profile.rqs - a list of high frequency profiles to be outputted (unit #13)

inhts.cfs grid, bathymetry, climatological fields (unit #14)

eta.flx - interpolated meteorological fields from the Eta model (unit #16);

run-dir/rs.current/YYMMHHDD.rs - input restart file (unit #17)

params - NAMELIST file for model parameters (unit #19)

prfxtr - NAMELIST file for high output frequency profile handlers (unit #21)

flux.avg or *flux.avg.most.recent* - spatially and temporally averaged fluxes (unit #52)

Some control information is passed to the model by a namelist file *params*. Model code is then compiled for every individual run. These parameters are:

MODE - 4 start with diagnostic, then with prognostic calculations
3 prognostic calculation
NREAD - 0 start from input climatology
1 start from restart file
INTRV - time interval of meteorological input (hours)
IPRINT - printout interval (hours)
LSAVE - sea elevation saving period at tide stations (interval in hours)
ISAVE - model averaged fields (E,T,S,V) saving interval (hours)
JSAVE - model instantaneous fields saving interval (hours)
KSAVE - high output frequency/surface fields saving interval (hours)
IDAYS - total integration time (days)
hh,dd,mm,yy - initial date and time (all real)

Control information regarding high output frequency profiles is passed to the model via a namelist file *prfxtr*. There are two parameters:

IPRFSAV - period of outputting (hours)
PRFTML - a Fortran format string that provides a template name for profile files. The format must include 2 integer specifiers to accomodate (i, j) indexes of given gridpoints that will be encoded into file names.

Output of the model falls into following categories:

YYMMHHDD-YYMMHHDD.avg - averaged model output fields (unit 90)
YYMMHHDD-YYMMHHDD.3D - instantaneous model output fields (unit 91)
YYMMHHDD-YYMMHHDD.mnx - extreme values of selected model fields (unit 92)
YYMMHHDD-YYMMHHDD.2D - high output frequency/surface fields (unit 93)
YYMMHHDD-YYMMHHDD.grd - grid and other fixed data (unit 94)
YYMMHHDD-YYMMHHDD.el - sea elevation at tide stations (unit 85)
YYMMHHDD-YYMMHHDD.tnc - SST at selected sites (unit 81)
YYMMHHDD-YYMMHHDD.tnu - SST at selected sites (unit 82)
YYMMHHDD-YYMMHHDD.tmd - SST at selected sites (unit 83)
YYMMHHDD-YYMMHHDD.tsn - SST at selected sites (unit 84)
YYMMHHDD-YYMMHHDD.dmp - redirected printouts (stdout)
YYMMHHDD-YYMMHHDD.pNN - high output frequency profiles (units #61-80)

All the above files are moved to directories branching from the *ao* subdirectory in the main run directory.

run-dir/rs.current/RS.YYMMHHDD - output restart file (unit 97)

4.5 Post-processing programs

The post-processing has been spread out between a number of programs due to the diverse nature of information being processed, and also for some convenience reasons in file handling. The general post-processing scheme is presented in Figure 2.

4.5.1 GRIB-coder for instantaneous fields

Purpose: Coding instantaneous fields into GRIB

Authors: C. Peters, L. Lobocki

Usage: *gribinst.x <3D-file> <gridfile> <output-gribfile> <diagfile>*

<3D-file> is the specification of the input file containing instantaneous, 3D fields to be coded

<gridfile> is the specification of the input file containing constant grid information (not shown in Figure 2)

<output-gribfile> is the specification of the output file containing GRIB-messages to be created

<diagfile> is the specification of the output file collecting diagnostic messages

Default location: /ombptmp/cfspom/CFS3.1/post/gribinst.x

Source code: /ombptmp/cfspom/CFS3.1/post/gribinst.f

Machine: Cray Y-MP series

4.5.2 GRIB-coder for averaged fields

Purpose: Coding averaged and extreme value fields into GRIB

Authors: C.Peters, L. Lobocki

Usage: *gribavg.x <avg-file> <gridfile> <mxmnfile> <output-gribfile> <diagfile>*

<avg-file> is the specification of the input file containing averaged, 3D fields to be coded

<gridfile> is the specification of the input file containing constant grid information (not shown in Figure 2)

<mxmnfile> is the specification of the input file containing fields of daily extremas

<output-gribfile> is the specification of the output file containing GRIB-messages to be created

<diagfile> is the specification of the output file collecting diagnostic messages

Default location: /ombptmp/cfspom/CFS3.1/post/gribavg.x

Source code: /ombptmp/cfspom/CFS3.1/post/gribavg.f

Machine: Cray Y-MP series

4.5.3 GRIB-coder for high output frequency (surface) fields

Purpose: Coding high output frequency (surface) fields into GRIB

Authors: C.Peters, L. Lobocki

Usage: *gribsfc.x <avg-file> <gridfile> <mxmnfile> <output-gribfile> <diagfile>*

<sfc-file> is the specification of the input file containing fields to be coded

<gridfile> is the specification of the input file containing constant grid information (not shown in Figure 2)

<output-gribfile> is the specification of the output file containing GRIB-messages to be created

<diagfile> is the specification of the output file collecting diagnostic messages

Default location: /ombptmp/cfspom/CFS3.1/post/gribsfc.x

Source code: /ombptmp/cfspom/CFS3.1/post/gribsfc.f

Machine: Cray Y-MP series

4.5.4 Profile and cross-section extractor

Purpose: Extracts selected cross-sections and profiles, including open boundaries, river mouths, and ship tracks. Output files may contain one or many profiles from multiple time steps.

Authors: L. Lobocki, J. Kelley

Usage: *prfxtrct.x <3D-file> <gridfile> <template> <YYMMDDHH> <start> <end> <step>*

<3D-file> is the specification of the input file containing instantaneous, 3D fields

<gridfile> is the specification of the input file containing constant grid information (not shown in Figure 2)

<template> is the template for the main part of names of the files that will be created. The program will append 2-3 character name extensions to the template provided for individual output files.

<YYMMDDHH> is the initial date for the run from which the processed results originate

<start> is the forecast hour of the first data to be outputted

<end> is the forecast hour of the last data to be outputted

<step> is the time period between two subsequent data packs, in hours.

Default location: /ombptmp/cfspom/CFS3.1/post/prfxtrct.x

Source code: /ombptmp/cfspom/CFS3.1/post/prfxtrct.f

Machine: Cray Y-MP series

Note: This program expects to read two data files, *oleander.ij* and *shngstar.ij* containing loactions of gridpoints along ships' routes. These files must be present in the current directory while executing the program.

4.5.5 Profile processor

Purpose: Processes the high-frequency profile output available at selected sites. Arranges data in time series and converts them into IEEE-format, readable by PC-based Profile Exploration Toolkit (PET).

Author: L. Lobocki

Usage: *prfgathr.x <YYMMDD1> <YYMMDD2>*

<YYMMDD1> specifies the beginning of the period from which the data are to be processed

<YYMMDD2> specifies the end of that period

Default location: /ombptmp/cfspom/CFS3.1/post/prfgathr.x

Source code: /ombptmp/cfspom/CFS3.1/post/prfgathr.f

Machine: Cray Y-MP series

Notes: This utility is not included in the typical execution suite, as the period of interest is usually longer than a single run length. At present, this program is run manually once a desired amount of results accumulates.

The program expects to find binary profile files as created by the POM code, in the current runtime directory (files, with their original naming are to be copied into a common directory from all *profile* branches of the directory tree branching from the *ao* subdirectory). Also, a copy of the *profile.rqs* file that has been used for the model run must be present in the current directory while running this utility. On output, converted profile files are saved as *pIIIxJJJ.prf*, where *III* and *JJJ* stand for i and j indexes of a given gridpoint. These files may be copied to the PC (use binary transfer option) and displayed using PET.

4.5.6 GRIB-Horizon converter utility

Purpose: to convert GRIB-coded Eta fields to a format, utilized by *Horizon*, a MS-Windows based, interactive data analysis and visualization system.

Author: L. Lobocki

Usage: documented separatedly (see author for details).

Default location (converter): /ombptmp/cfspom/tools/horzhook/cfs/cfsgb2hr.f

Machine: Cray Y-MP series

4.6 Batch mode graphics

All the batch mode graphics currently in use is based on workstation-based IDL software package. Individual plots utilize either 3D fields stored in the run directory, or preprocessed cross-sections stored in ao/YYMM/Xsect branching from the run directory.

4.7 Auxillary utilities

4.7.1 GRIB inventory utility

Purpose: to produce a readable listing of Product Description Sections (PDS) of GRIB messages contained in a GRIB conglomerate file.

Author: R.E. Jones

Usage: `invindex.x` *GRIB-index-file-name*

Default location - executables: /ombptmp/cfspom/tools/invindex.x

Comments: lists to stdout. Redirect output if needed.

Source code: /ombptmp/cfspom/tools/invindex.f

Source language: FORTRAN.

Machine: Cray Y-MP series

4.7.2 Surface file data inventory

Purpose: to list inventory of the POM surface input file

Author: L. Lobocki

Usage: `inveflx.x` *file-name*

Source code: /ombptmp/cfspom/tools/inveflx.f

Source language: Fortran 77

Machine: Cray Y-MP series.

5. Data files

5.1 Archived atmospheric surface fields (Eta forecasts)

The on-line archive is kept in */dm/cfspom/CFS/archive/etasrf*; individual files are placed in subdirectories denoted by year and month, *YYMM*. Data file names follow convention *YYMMDDsfRR.grb*, where *YYMMDD* stands for date, and *RR* for the Eta model resolution. Individual files contain daily sequences of GRIB messages as described in Office Note 388 (Stackpole, 1994), originating from both 00Z and 12Z forecast cycles, and stored every 3 hrs. throughout the forecast period. GRIB messages are placed sequentially, one after another, with an ASCII 'GRIB' indicator at the beginning of each message. The list of stored fields is shown in Table 5-1.

Table 5-1. Fields stored in the COFS atmospheric input archive.

Field description	Surface ID	Quantity ID
u-component of the wind speed at 10 m, along the model X-axis orientation	10 m	U GRD
u-component of the wind speed at 10 m, along the model X-axis orientation	10 m	V GRD
mean sea level pressure	SFC	PRES
air potential temperature at 10 m	10 m	POT
specific humidity at 10 m	10 m	SPF H
accumulated precipitation	SFC	A PCP
surface friction velocity, absolute value	SFC	FRICV
surface sensible heat flux	SFC	SHTFL
surface latent heat flux	SFC	LHTFL
net shortwave radiation at the surface	SFC	NSWRS
downward longwave radiation at the surface	SFC	DLWRF
sea surface temperature	SFC	WTMP
land/sea mask[2]	SFC	LAND

[2] It was found in early 1996 that the land mask stored in the Eta output was a combination of an ice mask and a land mask rather than a land mask alone. This caused the interpolation code to fail, and the code was temporarily adjusted to use older land mask data rather than obtaining the actual land mask every run anew.

5.2 Eta land/sea mask and grid files

The Eta land/sea mask file *smask.N* is created from the land/sea mask record found in the atmospheric surface fields file. It is saved as a single record containing sea mask array (following "filled i-j" indexing convention) in an integer binary format. Values are: 1 for sea, 0 for land.

The Eta grid files *gridNN.bin* contain pre-computed geographic longitudes and latitudes of individual Eta gridpoints. It is created whenever the interpolation program fails to find this file, or otherwise used as an input file.

intNN.bin files contain pre-computed indexes of Eta gridpoints assigned to individual COFS gridpoints for interpolation, Eta mass/velocity point flags and arrays of transformation coefficients needed to convert (rotate) vectors from Eta coordinate systems to COFS coordinates. Similarly to *gridNN.bin* and *smask.N*, *intNN.bin* is either read or created when missing.

5.3 Area-averaged flux database

flux.log is a direct access, constant-record length, ASCII file containing subsequent values of spatially-averaged fluxes for the entire COFS domain. Each record contains the following data, written in FORTRAN format (5(I3.2),10(1X,F8.3)):

year, month, day, forecast cycle, forecast time, net shortwave radiation at the surface, downward longwave radiation at the surface, surface sensible and latent heat fluxes as recalculated by the ocean model, surface sensible and latent heat fluxes as calculated by the Eta model.

Starting from version 3.0 of the COFS, this file is used only for diagnostic purposes. Prior to that version, daily averages of area-averaged radiative fluxes were used in the flux-correction scheme.

5.4 List of required GRIB messages

This file contains a list of GRIB messages as required to form the model input, forming a control file for the GRIB scan utility (section 4.2.2). The file is ASCII-formatted and has the following structure:

first line: comment - a ruler for the PDS vector

second line: PDS template vector: individual entries in JPDS/KPDS array as used by the NCEP's W3FI63 GRIB unpacker routine (refer to the documentation of W3FI63 routine for explanation of those; a list is also included below); a -1 value means a wildcard entry.

(1) id of originating weather forecast center

(2) generating process id number

(3) grid definition

(4) GDS/BMS flag (right adjusted copy of octet 8)

(5) indicator of parameter

(6) type of level

(7) height/pressure , etc., of a level
(8) year including (century-1)
(9) month ofyear
(10) day of month
(11) hour of day
(12) minute of hour
(13) indicator of forecast time unit
(14) time range 1
(15) time range 2
(16) time range flag
(17) number included in average
(18) version number of GRIB specification
(19) version number of parameter table
(20) nr missing from average/accumulation
(21) century of reference time of data
(22) units decimal scale factor
(23) subcenter number
(24-25) - reserved for future use

All subsequent lines, up to the terminator line, contain four numbers which override entries #5, 6, 7 and 16 of the PDS template vector. The date/time entries in the PDS are substituted with values decoded by the program from its command line parameters. Each of these PDS vectors is used as a template for searching GRIB messages with a matching PDS section in the GRIB index file.

The terminator line: "/" (slash) sign in the first column marks the end of the list. All subsequent lines will be ignored.

5.5 Atmospheric data files for the model execution

Consists of a series of record groups, written every 3 hrs. from model, in a binary format. Record groups contain: forecast cycle and time label, net shortwave radiative flux, downward longwave ratiative flux, air potential temperature at 10 m, u and v components of the velocity (oriented with respect to the ocean model grid), air humidity at 10 m, surface pressure and average precipitation rate for preceding 3-hr period.

For more details on units being applied and definitions, see comments placed in the SURF_ETA routine of the model.

6. Output archive

The output archive usually branches as *ao* subdirectory from the run directory. It is organized into monthly parts (Figure 3). Individual branches of the output archive are created on demand by the control program.

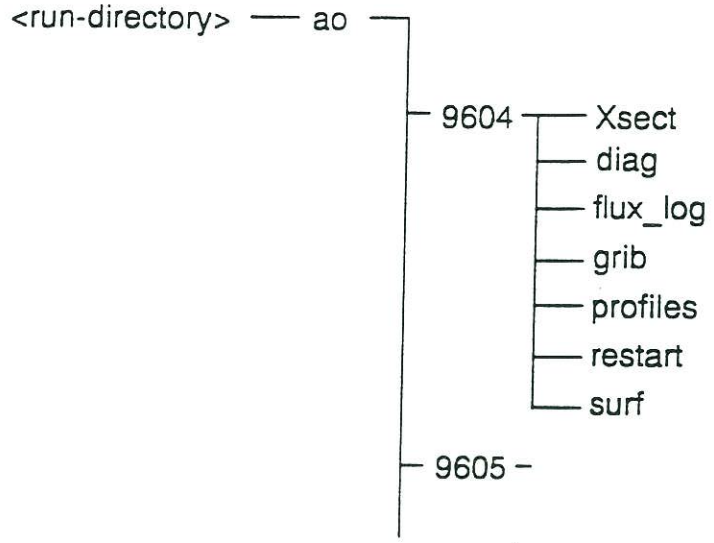


Figure 3. A schematic representation of an element of the COFS output directory structure.

Within every monthly branch, the following subdirectories are present:

diag - contains textual information outputted by model and other software components. Useful mainly for tracing purposes.

flux_log - contains subsequent copies of the *flux.log* files (5.3)

grib - contains various forms of GRIB-coded model fields: *.agb* files contain 24-hr averaged fields, and surface fields of daily extremes; *.igb* files contain instantaneous fields; high-frequency surface output is directed to *.sgb* fields.

profiles - contains profile repository. Files stored there are subject to further postprocessing spanning across multiple runs.

restart - contains restart files. Since these files are large, only some of the restart files are stored. In production mode, two restart files per month are typically retained.

surf - contains time series of hourly sea elevations and SST at selected measurement sites.

Xsect - contains cross-sections as generated by the profile/X-section extraction program.

7. The control program reference

7.1 Main program unit *(main; run31.x, run31.c)*

7.1.1 Purpose

The *main* unit controls the conversion and streaming of the continuous CFS-POM atmospheric input from the surface field files stored in the COFS input archive, by generating a series of UNIX shell, GRIB manipulation programs, and interpolation program calls. The unit also handles generation and submittal of a batch script controlling execution of the model and processing its output.

7.1.2 Invoking the program (command-line parameters)

The command line has the form

<executable-program-name> <ndays> <mode> <resolution>

where

<ndays> run extent in days,

<mode> may be either PROD (production, operational, forecast) or HIND (hindcast; see section 6 for explanation of operational and hindcast modes).

<resolution> Eta model resolution in km, such as 80, 48, 29

Command-line parameters are positional; when omitted, ***ndays*** defaults to 1, and *mode* is taken as HIND.

7.1.3 Return values

The program produces a return code through calls to the *terminate* routine (7.7.5), which handles certain execution monitoring operations and terminates execution by a call to system routine *exit*. The values returned upon termination are:

0 - normal termination

Fatal errors:

1 - wrong command line parameter(s)

2 - critical file system error

3 - the program was unable to establish the run duration

4 - necessary data were not found

5 - fatal error when executing an application or command

Warnings & cautions:

-1 - production run progressed with inconsistently dated restart file

-2 - no command-line parameters present, one day run duration assumed

-3 - failed to back up the area-averaged flux database

7.1.4 Settings of constants

Name of a constant	Type	Meaning/role
run_dir	char *	Path name of the main run directory
work_dir	char *	Path name of the temporary working directory
trace_dir	char *	Path name of the trace directory
rst_cur_path	char *	Path name of the current restart file directory
rst_prev_path	char *	Path name of the previous restart file directory
arc_dir_fmt	char *	C format string for encoding a path name of individual input archive subdirectories; year yy and month mm are to be encoded
gribix_name_fmt	char *	C format string for encoding a file name of individual archived GRIB index files within a subdirectory (path not included). Year yy, month mm, day dd of a forecast cycle, and model resolution, rr are to be encoded.
grib_name_fmt	char *	C format string for encoding a file name of individual archived GRIB files within a subdirectory (path not included). Year yy, month mm, day dd of a forecast cycle, and model resolution, rr are to be encoded.
rst_name_fmt	char *	C format string for encoding a file name of a POM restart file within a subdirectory (path not included). Year yy, month mm, day dd, hour hh of a forecast cycle, and the forecast time, tt are to be encoded.
res_dir_fmt	char *	C format string for encoding a path name of individual results archive subdirectories; year yy, month mm are to be encoded, and a proper name of a directory branch related to the type of output is to be appended
prog_name	char *	File name of the executable of the control program.
utl_spec_gribindex	char *	Full path name of the GRIB indexing utility (4.2.1)
utl_spec_inqgrib	char *	Full path name of the GRIB scanning utility (4.2.2)
utl_spec_intp	char *	Full path name of the interpolation program (4.3)
utl_spec_gribinst	char *	Full path name of the GRIB coder (4.5.1)
utl_spec_gribavg	char *	Full path name of the GRIB coder (4.5.2)
utl_spec_gribsfc	char *	Full path name of the GRIB coder (4.5.3)
utl_spec_prfxtrct	char *	Full path name of the profile/Xsection extracting program (4.5.4)
utl_spec_horzcnvt	char *	Full path name of the Horizon converter (4.5.6)
model_source	char *	File name of the model source code (4.4)

model_executable	char *	File name of the model executable code
model_include	char *	File name of the inclusion file to the model source code
ctl_spec_inqgrib	char * [3]	Full pathnames of control files for the GRIB scan utility (5.4), for individual Eta versions (80, 48, 29 km)
ctl_spec_horzcnvt	char *	Full pathname of a control file for the GRIB-Horizon converter
monthly_clim_bc	char *	File name of a file containing monthly climatological boundary conditions
initial_clim	char *	File name of a file containing grid, bathymetry, climatological fields
mean_bndry_elev	char *	File name of a file containing mean boundary elevations
bndry_tide_params	char *	File name of a file containing tide parameters at boundaries
profile_list	char *	File name of a file containing a list of high frequency profiles to be outputted
output_fname	char *	File name of a file containing interpolated meteorological fields from the Eta model
custodian	char *	An user name, or a list of user names of the user(s) who will receive job reports via e-mail. Individual entries may take a form of *user@machine-adress* where supported by local e-mail systems.
max_forecast_time_ stored	int [3]	A forecast time of the last Eta forecast to be retrieved, for individual Eta model versions
min_forecast_time_ stored	int [3]	A forecast time of the first Eta forecast to be retrieved, for individual Eta model versions
time_storing_ increment	int [3]	Time increment for searching thru daily GRIB files, for individual Eta model versions
cutoff_time	int [3]	In Eta-Meso, the assimilation time included in the forecast period, for individual Eta model versions.
Eta_resolution	int [3]	Eta model resolution [km]. Used to identify files.
NMC_grid_codes	int [3]	NCEP ON-388 codes for individual Eta grids.
model_preference_ table	int [3]	A table defining the hierarchy of preferred Eta model versions.
cycle_priority_table	int [3][3][9]	Forecast cycle priority table
ftime_priority_table	int [3][3][9]	Forecast time priority table

7.1.5 Program execution

The program takes the following actions:

(1) The trace file subsystem is arranged and initialized

(2) Working directory is established

(3) *RequestAnalysis* module is called to perform a preliminary determination of the the first forecast cycle to be used, run duration and mode (see section 7.2).

(4) Control and data files are brought to a temporary location, when necessary

(5) A daily loop is executed, stepping through the expected ocean model run duration. Inside this loop, an internal time loop is executed, starting from the first forecast cycle as determined by *RequestAnalysis* for the first day and from 00Z for all subsequent days, and proceeds through the run duration with a time increment as specified by a constant time-storing increment (see 7.1.4).

Within this internal loop, the presence of data is checked by a call to *data_found* (7.3), and the interpolation code *utl_spec_intp* is executed (7.1.4; 4.3), either overwritting the output data file (7.1.4; 5.5) in the first cycle, or appending it.

If the data for a given time are not found, the program checks if there are other data available which could be used in place of missing data- the check involves three adjacent forecast cycles of the Eta model in order of preference as described in priority tables *cycle_priority_table* and *ftime_priority_table* (7.1.4; see Figure 4-1). If data are found, the program uses them to append to the ocean model input; if the program finds it impossible to patch the gap, the run is aborted.

The check is made separately for all subsequent forecast times.

(6) A batch script is generated to control model execution. At present, many of the present data file locations are hardwired in this part of the code.

(7) Batch job is submitted for execution. After executing the model, postprocessing will be carried out by the batch job, and (exclusively for production runs) the batch job will launch another batch job controlling dissemination of the products. Finally, for hindcast runs longer than one day, the batch job will execute the next run of the control program, which provides automatic chaining.

(8) Finally, the program calls the *terminate* routine (7.7.5) to close trace files, handle e-mail notification, and to stop the program.

7.2 Request analysis (RequestAnalysis)

7.2.1 Purpose

The *RequestAnalysis* module preliminarily identifies the starting date, operation mode, and run duration. The decisions made at this stage depend solely on the user requested parameters, current date and the restart file date, and not on the actual availability of the atmospheric data, whose availability is checked elsewhere.

7.2.2 Synopsis

int RequestAnalysis (int argc, char *argv[], char *s_forecast_cycle,
int *prod, date *first_forecast_cycle,
date *today_date, int *idays, int *run-length)

int argc - number of command-line arguments to be parsed

*char *argv[]* - array of pointers to individual command-line arguments

*char *s_forecast_cycle* - pointer to a character string to receive the restart file date as read from the restart file name, as a form of *yymmddhh* string

*int *prod* - points to an integer variable to receive operational mode flag

*date *first forecast-cycle* - points to a date structure to receive the date record as determined from the starting forecast cycle

*date *today_date* - points to a date structure to receive the current date

idays - points to an integer variable to receive run duration in days

*int *run-length* - points to an integer variable to receive run duration in hours

Returns:

0 - normal termination

Fatal errors:

1 - unable to find directory

2 - critical filesystem error

3 - failed to establish the run duration

Warnings:

-3 - ignoring the request in an operational mode because of restart file being up to date

7.2.3 Function

The *RequestAnalysis* function takes the following actions:

(1) checks the presence of individual directories. In a case of missing or incorrectly named directories which should contain source data, control is passed to the error conditions handler. In case of missing working directories, the system will attempt to create them.

(2) parses the command line

(3) locates the current restart file and decodes the time/date stamp from its name.

(4) rejects or ignores requests when the the restart file is up to date.
In the operational mode **the system is run twice a day**, following completion of nominal 00Z and 12Z cycles of the Eta-MESO model. However, under normal schedule, the COFS forecast is initiated only during 00Z cycle. In certain situations, the available Eta forecast may not suffice to continue the run throughout 24 hrs. (see Figure 4-1), and the execution is shortened to 12 hrs. The sole purpose of the 12Z run is to extend such a shortened 00 forecasts to 24 hrs. when the needed data arrive. Therefore, in the production mode the system finds the restart files up to date in most 12Z runs, and error conditions do not arise.

In the hindcast mode, the presence of an up-to date restart file is treated as an error, and the request is rejected.

(5) determines preliminarily run duration.

In the operational mode this duration can be either 24 or 12 hrs., with the latter chosen only for runs starting at 12Z. In the hindcast mode, runs extend to the demanded number of days as specified by the first command-line parameter, unless that would extend beyond the current 00Z+24 limit. In the latter case, the run duration is reduced.

During this phase, no attempt is made to determine whether all the required data are present or not, so further modifications of the run extent are possible at later stages. In normal conditions, the 00Z run performs a 24 hr forecast, and the 12Z run is skipped. However, when input data available at that time do not cover the +12 to +24-hr. range, the 00Z run is shortened to a 12-hr forecast, and extended later (by a 12Z run) to 24 hr when the next Eta forecast arrives.

7.3 Data presence inquiry (data_found)

7.3.1 Purpose

Controls data availability inquiries done through *utl_spec inqgrib* (7.1.4; 4.2.2). Organizes bringing GRIB and GRIB index files from the archive to a temporary location and/or generation of GRIB index files when necessary.

7.3.2 Synopsis

int data found (date cycle, int ftime, char *grib_name, char *gribix_name)

date cycle - a date structure saving the forecast cycle to inquire about

int ftime - forecast time to be inquired about

*char *grib_name* - a pointer to a character string containing filename of the GRIB file to be scanned

*char *gribix-name* - a pointer to a character string containing filename of the GRIB index file to be scanned

Returns: 1 - when data were found, 0 - otherwise (may include problems with accessing an existing file)

7.3.3 Function

data_found checks the availability of required GRIB/GRIB index files in the temporary directory. In case of their absence, or when the archived files appear to be newer than their copies in the temporary directory, an attempt is made to copy the files from the archive to the temporary location, and/or to create GRIB index files using *utl_spec_gribindex* (7.1.4; 4.2.1) when necessary. Next, GRIB index files are searched through using *utl_spec_ inqgrib* (7.1.4; 4.2.2) to find whether all the data specified by the control file *ctl_spec_inqgrib* (6.1.4; 5.4) are available.

7.4 System interface routines (run, prepdir, FileLastModified, FileSize, FileIsMigrated, FileExist, pcFileLastModified, SmartCopy)

7.4.1 run

Purpose: Passes a command indicated by the string *s* to the execution by the shell, echoing the command to trace files at tracing levels 3 and higher. Waits for the spawned process to terminate, and interprets the exit code.

Synopsis: int run (char *s)

Returns: 0 when the exit status of the process was 0, or the value returned by the process plus 100, when the process returned with an exit code. Otherwise, 5 when the command execution terminated due to a signal, or when it was stopped.

Notes: writes error messages to trace file system when necessary.

7.4.2 prepdir

Purpose: function attempts to make a directory.

Synopsis: int prepdir (char *dir, int ErrProc)

dir points to a character string containing path name of a directory to create.

ErrProc should be set to 0 when the error-tracing system is ready for operation.

Returns: 0 when the directory was succesfully created or already exists, 1 otherwise.

Notes: writes error messages to trace file system when necessary.

7.4.3 FileLastModified

Purpose: retrieve the time/date of last modification of a file identified by **path*.

Synopsis: time_t FileLastModified (char *path)

Notes: writes error messages to trace file system when necessary.

7.4.4 FileSize

Purpose: retrieve the size of a file identified by *path.

Synopsis: off_t FileSize (char *path)

Notes: writes error messages to trace file system when necessary. Returns -1 when the file identified by **path* does not exist, or **path* is not a valid name of an existing, accessible file.

7.4.5 FileIsMigrated

Purpose: finds whether the file identified by **path* is migrated or not.

Synopsis: int FileIsMigrated (char *path)

Notes: writes error messages to trace file system when necessary. Returns -1 when the file identified by **path* does not exist, or **path* is not a valid name of an accessible file.

7.4.6 FileExist

Purpose: finds whether the file identified by **path* exists and is not empty.

Synopsis: int FileExist (char *path)

7.4.7 pcFileLastModified

Purpose: retrieve the time/date of last modification of a file identified by **path*.

Synopsis: char *pcFileLastModified (char *path)

Notes: returns a pointer to a string containing the date/time.

7.4.8 SmartCopy

Purpose: function attempts to copy a file pointed to by *name* from the directory path pointed to by *from* to the directory path pointed to by *to* in the following situations:

(1) The file is not present on a target path

(2) Target file is empty

(3) Target file bears modification stamp older than the source

(4) Target file is shorter than the source.

Upon completion of copying, the presence of a file in target directory is verified. Error messages are posted when the source file is not present, when the copy fails or where target file is still not present. Copy commands are echoed to the full trace file.

Synopsis: int SmartCopy (char *name, char *from, char *to)

Returns: 0 when the target file was up to date, or if it was succesfufly copied, or a non-zero value otherwise.

7.5 Calendar routines (day_of_year, YYDDD, ParseDate, IncDate, DecDate, fix_date, DateString)

7.5.1 day of year

Purpose: returns the number of given day **d* in a year.

Synopsis: int day_of_year (date *d)

7.5.2 YYDDD

Purpose: returns the number of given day **d* in a year, plus 1000 times the current year in a century.

Synopsis: long int YYDDD (date *d)

7.5.3 ParseDate

Purpose: decodes a date/time **d* given in a character string pointed by **s* as *yymmddhh*

Synopsis: int ParseDate (char *s, date *d)

Returns: 0 when date succesfully decoded, 1 otherwise

7.5.4 IncDate

Purpose: Increments a date/time *d* by a given number of hours, *nhours*.

Synopsis: IncDate (date *d, int nhours)

7.5.5 DecDate

Purpose: Decrements a date/time *d* by a given number of hours, *nhours*.

Synopsis: DecDate (date *d, int nhours)

7.5.6 fix_date

Purpose: Corrects date/time records when the hour field contains a value smaller than 0 or greater than or equal to 24.

Synopsis: fix-date (date *d)

7.5.7 DateString

Purpose: Provides a textual representation of a date/time record *Date*

Synopsis: char *DateString (date Date, char *buffer)

Note: *buffer* must point to a character array capable of containing the date string.

7.5.8 CopyTime

Purpose: copies a date record *source* to the location pointed to by *target.*

Synopsis: CopyTime (date *target, date source)

7.5.9 CopyTime

Purpose: establishes if a given year is a leap year.

Synopsis: IsLeapYear (int year)

Returns: 1 when leap, 0 otherwise

7.6 Auxiliary routines and macros (imin)

7.6.1 imin

Purpose: Returns smaller of the two given integers *i1*, *i2*

Synopsis: int imin (int i1, int i2)

7.7 Execution monitoring system (OpenTraceFiles, PrintTrace, RenameTraceFiles, terminate)

7.7.1 An overview

The control program maintains a hierarchical system designed to provide convenient run execution monitoring and tracing capabilities. Messages such as error notifications, warnings, cautions, and tracing information are written to a five-level trace file system.

The level 0 is reserved for critical errors, which cause the run to fail. Level 1 trace files contain a short report, which consist of the request report, warnings and a job conclusion. Level 2 contains an extended report which adds reports of individual data searches operations. At level 3, (short trace), all major program operations are traced. Finally, level 4 (full trace) contains detailed tracing information which may originate from any level of the program structure.

Upon job termination, trace files are written into directory *trace_dir* (7.1.4) and labeled with a job run date. The system has an e-mail notification feature, which enables any of the trace files to be mailed to the user upon job completion. This feature is controlled by the *TraceLevelWanted* constant and user name(s) specified in the character constant *custodian* (7.1.4).

7.7.2 OpenTraceFiles

Purpose: Opens a global array of trace files located in a trace directory *trace_dir* (7.1.4). The template name of trace files is set inside the routine body.

Synopsis: int OpenTraceFiles ()

Returns: 0 when successful, 2 otherwise.

Discussion: Opening trace files is meant to be one of the first operations of the program. At this moment, the data needed for convenient identification of these files are not available. Thus, the names given to the trace files are to be treated as temporary. Trace files are renamed upon job completion, unless the control program execution terminates prematurely. On these rare occasions, trace files remain under their temporary names and may be overwritten by subsequent runs. An emergency mail message is sent to the users specified by *custodian* (7.1.4) when the routine fails to open trace files.

7.7.3 PrintTrace

Purpose: posts a message pointed to by *msg* into all trace files of level *j* and higher.

Synopsis: void PrintTrace (int j, char *msg)

7.7.4 RenameTraceFiles

Purpose: renames trace files by appending the date string pointed to by *datestring* to their temporary names, followed by a period (.).

Synopsis: int RenameTraceFiles (char *datestring)

Returns: 0 when successful, 2 otherwise.

Notes: An emergency mail message is sent to the users specified by *custodian* (7.1.4) when the routine fails to open trace files.

7.7.5 terminate

Purpose: handles trace file closing, mail notification, and terminates program via call to *exit*, passing *iret* as the exit code.

Synopsis: void terminate (int iret)

Return: none. A call to *terminate* stops the program execution.

8. Installation and operation instructions

8.1 Prerequisites & portability issues

The system has been developed and tested on a CRAY C-90 series computer, running under the UNICOS 8.0 operating system. The system design assumed a certain hardware and software configuration of the machine to be used, and a certain setup of the production system. As such, it may need modification when brought to other systems. In particular, the system assumes existence of an on-line Eta surface data archive or its functional equivalent. The archive organization and operation is documented separately.

The control program was written in C language, and compiled using Cray's standard C compiler, c89. System-level calls, which may be UNICOS-specific, were encapsulated in subroutines to provide an easier way of implementing on other systems. The components' code was written in FORTRAN-77 and compiled by Cray's cf77 FORTRAN compiler, using NCEP's "W3" library containing GRIB decoding routines and data format conversion support. A source code of W3 routines is available from NCEP along with data format conversion routines for various platforms. In general, users working on machines other than Cray's Y-MP/C-90 series should implement GRIB-decoding software before attempting implementation of components documented here.

In addition, components communicate with the control program via command line/return code mechanisms, which are supported by system-dependent library routines. Calls to these were encapsulated in separate modules to facilitate implementation on other systems.

Input data files used by the model include the *inhts.cfs* file, which contains grid definition, bathymetry, initial climatological fields; monthly climatological boundary conditions file *tbs_tbe.mon*; tide parameters at the boundaries file *tide6.bdr*; mean boundary elevations file *els_ele_.2yr*; and a restart file *YYMMDDHH.rs*, which contains all model variables required by the model initially.

8.2 Naming recommendations

The NCEP setup of the COFS utilizes a CD-R mastering device for storing model input and output archives, and also copies of the source codes and required data. This imposes limitations on file names that are permitted on CD-R's. Naming convention is standardized by the ISO-9660 norm which requires the file name to consist of a main part, containing up to 8 alphanumeric characters, followed by a period, and the extension part, where up to three alphanumeric characters are allowed. In addition, an underscore "_" may be also used in place of any alphanumeric character. Note that this standard is more restrictive than the convention used in popular MS-DOS systems, where other characters are also permitted. To avoid unnecessary name conversions while backing up the system to CD-R's and restoring it, we recommend using names consistent with ISO 9660 wherever possible and reasonable.

8.3 Source code updating recommendations

As both the model code and the control program code at present undergo frequent changes by different persons, it is highly recommended to follow certain code maintenance

procedures such as one introduced at NCEP for this purpose. Starting with every major update as a common reference code, incremental changes are introduced using conditional compilation directives, implemented in both C and FORTRAN compiling systems under UNICOS 8.0.

In particular, the following construction is useful:

#ifdef <symbol>

  <newly introduced part of code>

#else

  <old part of the code to be replaced by the new version>

#endif

The above may be placed in many places in the code so that related changes occuring in many places are automatically linked. To activate changes, a directive

#define <symbol>

must be placed in the same file, preceding all #ifdefs referring to the same symbol. The <symbol> can be any name, such as LL_960529a, where two first characters code initials of the author of the proposed modification, and a date of the modification follows. The scope of the #define directive extends throughout the entire file starting from the place where it was introduced.

The cf77 FORTRAN compilation system will call a 'generic preprocessor', gpp, when the file name of the code to be compiled ends with ".F" (a period, then capital F). This will cause directives to be evaluated in accordance with currently defined symbols; the gpp will produce a resulting 'clean' FORTRAN code (i.e., updated version with no preprocessing directives) and direct it to further stages of compilation. It is also possible to produce the 'clean' FORTRAN code out of the files containing directives, by using stand-alone gpp, e.g.:

gpp -P myprog.F > myprog.f

The conditional compilation features described above are included into the definition of standard C, so that they are supported by any C compiler system, not only Cray's c89.

8.4 Run installation checklist

To set up a new run, follow the steps listed below.

- ☐ establish a run directory
- ☐ create subdirectories: *rs.current*, *rs.previous*, *ao*, and *trace* within the run directory
- ☐ copy: model code (*pomcfs31.F*, *comblock.h* (ver. 3.0 - *pomCFS.f*, *comblk.h*)), the code of the control program (*run3l.c* (ver. 3.0 - *run30.c*)) into the run directory
- ☐ edit the control program code settings section, paying attention to constants describing location of data files and executables. At least, assign unique names using *run_dir*, *work_dir*, *rst cur_path*, *rst_prev_path* and *prog_name* constants along with all hardwired trace file names to provide unique identifiers for the run and to avoid conflicts with other runs.

- ☐ Make sure that all specifications (7.1.4) of control files (5.4) and executable components (4.2-4.5) point to the proper files existing on your system.
- ☐ If the components have not yet been installed on your system, compile them and link with library routines, and place executables in proper directories. Check section 7.1.4 and the settings section in the main unit of the control program for a full list of required files.
- ☐ Scan the code for possible former locations (elements of the former run directory path name) of system components and update locations when necessary.
- ☐ make sure that the symbol *dissemination_allowed* is not defined, unless you are setting the operational run to be executed in the forecast mode.
- ☐ compile the control program code,
- ☐ copy data files: *inhts.cfs, tbs_tbe.mon, els_ele_.2yr, tide6.bdr,* into the run directory
- ☐ copy data files, if available: *smask.1, smask.2, smask.3, grid94. bin, grid96 bin, int94.bin, int96.bin, grid90.bin, int90.bin,* into the run directory
- ☐ copy a restart file into the *rs.current* directory. The name of the restart file must conform to the template used by the program.
- ☐ The *rs.current* directory must not contain any restart file other than the current one.
- ☐ If the model is to be launched by other jobs, or initiated in batch mode, place the file *launcher.prod.qsub* in your run directory and edit it, creating appropriate directories and names.
- ☐ make a *post* subdirectory in the run directory, and place "*.*ij*" files in it, to be used by the profile/cross-section extraction postprocessing program.
- ☐ if the run is a continuation of another run and the flux monitoring database *flux.log* is to be continuously maintained, copy the file *flux.log* into the run directory. Otherwise, make sure that *flux.log* is not present when you initiate the run, either in run directory or in the temporary working directory.

8.5 Troubleshooting and emergency procedures

8.5.1 General troubleshooting

(1) If you have started the control program in interactive mode, try to retrieve the exit code immediately after termination of the program. Check section 7.1.3 for an explanation of possible reasons.

If you launched it from a batch script, echo the exit code within the script.

(2) Check mail messages. If you were receiving no messages, the mail locations are correct and the control program is supposed to be launched by another application verify if launching of the model actually took place.

(3) Check trace files in your trace directory. If the directory contains non-renamed files(without the date extension), a breakdown of the control program execution is apparent. At present, there is no experience gathered with such breakups. Otherwise, look for possible warnings in the trace files and try to trace execution flow with the source code.

(4) Check the contents of the working directory, if recent. If you cannot isolate the problem by checking the working directory which might have been 'overwritten' by several subsequent runs, delete all the files in the working directory and rerun the system again.

(5) If the control program passes successfully to launching the model run in a batch mode, but the model fails, check batch job reports left in the working directory (*cfs.qsu.o...*). Also, textual model output, directed to the *CFS.out* file in the working directory and copied on job completion to *.dmp* files in the *ao* subdirectory of the run directory may be useful in locating reasons for the failure.

8.5.2 Emergency procedures for the operational run

The operational run is launched twice a day, soon after the Eta output is archived. The archiving procedure, in turn, is initiated by the Eta model operational scripts. Occasionally, due to data unavailability, missed model launches, or for other reasons, the run may stop or become "late". The latter conditions arise when the current restart file becomes older than one day. As the control program is designed to handle both the current operational and the hindcast runs, the same code may be used for advancing the run in a hindcast mode up to the current date. It is necessary, however, to figure out whether the reason for the failure is no longer active, as discussed in section 8.5.1, and then to correct the problem, if necessary. You may run the control program in an interactive mode as well as in batch mode.

9. References

Blumberg A.F., and G.L. Mellor, 1987: A Description of a Three-Dimensional Coastal Ocean Circulation Model. In: *Three Dimensional Coastal Ocean Models*, [ed.: N.S. Heaps], 1-16.

Mellor G.L., 1996: User's Guide for a Three-Dimensional, Primitive Equation, Numerical Ocean Model. Manuscript, Princeton University, rev. June 1996.

O'Connor, W.P., 1991: A User's Manual for the Princeton Numerical Ocean Model. Rep. No. SP-5, Institute for Naval Oceanography.

Stackpole J.D., 1994: A Guide to GRIB. NCEP Office Note 388.

COFS input archive

description, operation and maintenance

Lech Lobocki
Environmental Modeling Center
Ocean Modeling Branch
May 20, 1996

1. Structure of the archive

The Coastal Ocean Forecast System (COFS) input archive consists of meteorological surface fields, generated by the operational mesometeorological model (the Eta model). These fields include:

- 10-m wind U-component
- 10-m wind V-component
- surface pressure
- 10-m potential temperature
- 10-m specific humidity
- 3-hr. accumulated precipitation
- surface friction velocity (abs. value)
- surface sensible heat flux
- surface latent heat flux
- surface net shortwave radiation
- downward longwave radiation flux
- sea surface temperature
- sea/land mask (1 record per cycle)

Currently (spring 1996), the source Eta model is the 29-km 'MESO-Eta'. This version runs operationally twice per day (00Z and 12Z cycle), delivering 36-hr. forecasts. At present, a 3-hr data assimilation cycle is included into this period, so that it effectively produces 33-hr. forecasts, starting 03 and 15Z. The surface fields are produced every 3 hours.

Other setups of the Eta model exist; the so-called 'Early' 48-km Eta runs twice per day producing 48-hr forecasts preceded by a 12-hr assimilation cycle (not included in the forecast period); former 80-km 'Early' Eta ran in a similar setup with except for assimilation, which was implemented later. At present, the Early Eta produces output every 6 hours.

The 29-km archive was established in July, 1995; 80-km archive existing prior to that date was converted to a compatible format (GRIB) and stored along with the new part. 48-km archive was also operating since introduction of the 48-km Eta model to the operations on October 12, 1995 (80-km version was retired on that day), but has not been used due to insufficient product frequency. A small archive containing wind analyses produced by 48-km Eta's data assimilation system (EDAS) was also in use in spring '96 for research purposes.

The main part of the archive - the 29-km Eta surface forecast archive - gathers data necessary for semi-operational COFS runs, parallel runs, and future hindcast runs. It collects 13 data sets per single Eta forecast run, totaling 157 records coded in GRIB (see Office Note 388). Per month, storage demand reaches 500 MB.

The archive consist of on-line and off-line parts. The off-line part is stored on CD-R disks and backed on CREEL tapes. The on-line part is stored on the Cray-3 (sn4021) disk assigned to OMB parallel runs, */ldiskb*, in subdirectories of the directory */ombptmp/cfspom/archives*. The directory structure is explained in Figure 1.

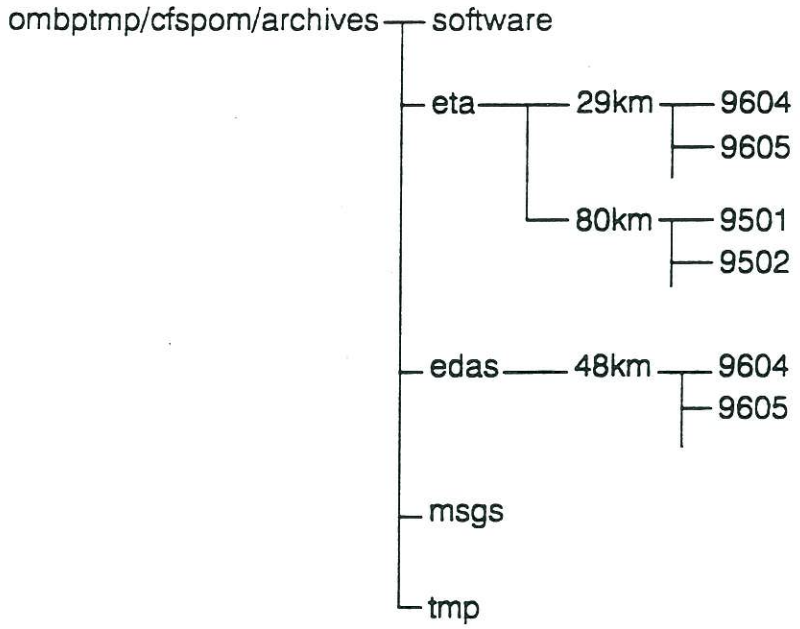


Figure 1. Structure of the archives directory.

The 48-km Eta forecast archive is currently stored on the DMF (*/dm/cfspom/archive...*) and has similar structure.

A quantum of data storage in the archive is a daily file, which gathers all the data originating from a given day. The file name follows the pattern *YYMMDDsfRR.grb*, where YYMMDD stands for the date, and RR is the Eta model resolution (80, 48 or 29).

2. Archive maintenance

The archive operation is critical to the COFS project. The COFS data feeding system in its present form can survive gaps in the input data no wider than two subsequent cycles of the Eta forecasts. For this reason, operation of the archives must be routinely supervised. In addition, data must be transferred periodically from on-line to off-line part, freeing disk space needed for subsequent writing. Transferred files may be brought back to the on-line part on demand.

The archive supervisor must check, on a daily basis, and make sure that:

- there is sufficient disk space for writing on the disk (Unix command *df*.)
- the archived files are being correctly written.

There are various ways to find out whether the archives operate correctly. Most easily, a correct operation of the 'operational' COFS run, which can be told by observing everydays' progress of its products or restart files is a symptom of correct operation. Note, however, that the COFS data management has an ability to patch some of the missing data automatically, so that forecast may still be executed even when fresh data stream has stopped. In these situations, warnings are issued, so watching the operational run may be sufficient. Next, there is a mail facility built into the archiving code, which directs the execution reports to the user. Again, in some special situations these reports may be misleading, or they may be confused with formerly sent reports. The third usual way to find out whether the operation of archive is correct is to produce inventory of archived files. Finally, tools exist to decode and plot the fields whenever necessary.

When the archive operations breaks for reasons other than lack of the Eta products, it is often possible to recover missing data and fix the archive within first few hours, when the Eta products still reside in rotating mass storage. A quick action is essential to restore proper operation.

3. Software

Archive procedures are activated automatically as batch jobs by the Eta model operational script (job #667). This job contains an Unix command
qsub -u cfspom $CFS
which sends an NQS request, passing the $CFS variable as a name of the run script. At present, this script is */ombptmp/cfspom/archives/software/archmeso.qsub* (Table 1).

```
#QSUB -lM 2Mw -lT 150 -x -eo -s /bin/sh
#QSUB -o /ombptmp/cfspom/archives/msgs/archmeso.rpt
cd /dm/cfspom/CFS/archive/msgs
cd /ombptmp/cfspom/archives/software
sh archmeso.sh ${PDY}$TM03 $COM `hostname` $ENVIR >> \
/ombptmp/cfspom/archives/msgs/archmeso.rpt
cd /ombptmp/cfspom/CFS3.0
qsub /ombptmp/cfspom/CFS3.0/launcher.prod.qsub
```

Table 1. The shell script *archmeso.qsub* controlling first stage of the archiving.

Note *-x* parameter of the *QSUB* command, which allows environmental variables to be passed from the calling script to *archmeso.qsub*. These variables, representing certain date/time and directory information are reformatted and passed out to the shell routine archmeso.sh, which continues processing. Also, *archmeso.qsub* initiates the execution of the 'operational' COFS run.

The *archmeso.sh* shell script, together with executable *archmeso.x* (source code *archsave.c*), controls the extraction of required data from the Eta output, and arranging

the data into files and directories. The *archmeso.sh* can be invoked manually; the following command syntax is required:
sh archmeso.sh <date> <dir> <hostname> <environment>
where <date> stands for the date in YYMMDDHH format, <dir> is the name of the directory where full Eta products are stored, <hostname> is the host name of the computer/datasystem these files are located on, and <environment> is a character string used to distinguish between production and non-production runs. The two latter parameters are not used in current *archmeso* implementation, but may be used in future extensions.

The archmeso system utilizes three W3LIB-based FORTRAN programs that can be used as standalone utilities. The *grbindex.x* produces an index file for GRIB collections produced by the Eta model. The responsive command syntax is:
grbindex.x <input-GRIB-file> <output-GRIB-index-to-be-created>
The *xtrgrib.x* utility scans the GRIB index file, locates desired GRIB messages fitting descriptions provided on the list, and extracts these messages from the GRIB file, placing them in the output file. The command line syntax is as follows:

xtrgrib.x <input-GRIB-file> <input-GRIB-index-file> <output-GRIB-file> <date>

The *xtrgrib.x* program takes a control file, *xtrgrib.ctl*, which is an ASCII text file, containing a list of desired GRIB messages. The structure of this file is explained in Table 2.

```
;1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 7 85 94 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1 -1 -1 -1 -1 -1 1 20 -1 -1 -1 -1
  33  105   10    0
  34  105   10    0
   1    1    0    0
  13  105   10    0
  51  105   10    0
  61    1    0    4
 253    1    0    0
 122    1    0    0
 121    1    0    0
 111    1    0    0
 205    1    0    0
  80    1    0    0
/
```

Table 2. An example of the *xtrgrib.ctl* control file. First line, starting with semicolon, is a comment text which provides numbering for JPDS items (see routine *GETGB.f* by M. Iredell). Second line (first non-comment line) contains a template for JPDS. All subsequent lines before the termination mark ("/" in the first column) contain JPDS items no. 5, 6, 7 and 16 (PDS octets 9, 10, 11-12 and 21, correspondingly).

These utility programs may be found in */ombptmp/cfspom/tools*. In addition, there is also inventory listing program *invindex.x* and a conversion utility *cfsgb2hr.x* which decodes the GRIB messages and writes IEEE-coded binary files, readable by a PC-based interactive graphic utility *Horizon*. The GRIB inventory program lists a GRIB index file as created by *grbindex.x* to the *stdout*, with the index file name passed as a command line parameter. The *Horizon* utility and its converters are documented separately.

4. Procedures

4.1. Moving data off-line: CD-R recording

This section describes a technique of moving the data from the on-line part to the off-line permanent storage based on CD-R, and is specific to OMB's hardware used for this purpose. The hardware is a PC-486 based workstation *lenox1*, equipped with Pinnacle RCD-1000 CD-R mastering device and a 1GB disk buffer, and running either *Linux* or *MS-DOS/MS-Windows 3.1* operating systems. A single CD-R may hold up to 620 MB of data; recording takes approximately an hour. Additionally, data must be transferred from the Cray disk through a network, and the disks should be verified after recording. Overall, the whole procedure lasts approximately 4 hours; use of *lenox1* must be also coordinated with other users.

Recording sessions must be run under *MS-Windows 3.1* while the normal operating mode for *lenox1* is Linux. Moreover, data transfer is significantly faster under *Linux*. For this reason, we include a brief description of booting procedures for both systems. Next, data transfer, and recording itself is described.

4.1.1. Booting procedures for linux1

If the machine is running the system which is not appropriate for the given step of the procedure, it must be taken down first. To do that,
in *Windows*: close all the tasks running, then go to the *Program Manager* and close it, either by selecting *Exit* from *File* menu, or by pressing Alt-F4 and confirming. When *Windows* shuts down, press Ctrl-Alt-Del.
in *Linux*: press Alt-F1 to go to the root window, then press Ctrl-Alt-Del and wait until the system shuts down and starts rebooting.
DO NOT reboot the machine by switching the power off or pressing the RESET button unless the system is out of control. This may result in garbage files cluttering the disk, or, in some cases, in damage of the file system.

While booting up, both the recording device and the external disk storage should be powered on. The default mode is *Linux*; to boot *DOS/MS-Windows*, wait until prompt "*LILO*" appears, and then quickly press left Ctrl key. The machine will respond with "*boot:*" prompt; type "*dos*" then. When the MS-DOS finishes booting, type "*win*" to start *MS-Windows*.

4.1.2. Data transfer

To facilitate operations, the directory structure on a CD-R is similar to the one in the on-line part, except for the initial */ombptmp/cfspom/archives* path. That is, the CD-R image contains a directory *eta*, under which there may be either *29km* or *80km* subdirectory, containing one or several monthly directories. The initial step of the data transfer procedure is to arrange a proper directory structure on a buffer disk. If the data are transferred under *MS-DOS/Windows*, the buffer disk is achievable as D:. Under *Linux*, the disk is seen as */export/dosd*. One needs to dismount and remount this drive to avoid file ownership problems if the data are to be imported under *Linux*. Next, either *DOS* commands *mkdir* and *cd*, *Linux* commands *md* and *cd*, or *Windows File Manager* utility may be used to arrange a proper directory tree on a buffer disk. It is not necessary to remove files already present on a buffer disk unless there is no sufficient space remaining. The CD recording software enables selection of files to be recorded at any directory level, including individual files.

Next, files to be recorded on a CD-R must have names conforming to the ISO9660 standard, that is, up to 8 alphanumeric characters followed optionally by a dot (.) and up to three-alphanumeric characters filename extension. Underscore character is also permitted either as a part of the main name or the extension. Thus, files in the archive must be renamed accordingly during the procedure.

To preserve name uniqueness, we omit "sf" characters in every filename while transferring files to the recording machine. The following Unix script copies archived files to a directory *../tmp* with proper renaming:

```
for i in *sf29.grb ; do
j=`echo $i | cut -c 1-6`
com="cp "$j"sf29.grb ../tmp/"$j"29.grb"
$com
done
```

To accomplish data transfer, various implementations of the *FTP* utility may be used. Data may be shipped out of the Cray by invoking *Unicos' ftp* utility, or imported from the Cray by invoking either *DOS*- or *Linux*-based *ftp* on the *lenox1*. In any case, non-ASCII data such as GRIB files must be transferred in *binary* (*image*) mode. Some *FTP* implementations default to ASCII transfer mode which results in adding text-formatting characters to form carriage return - line feed sequence used on some systems, including DOS. If the files are originally gathered in a single directory, a multiple get (*mget*) *FTP* command is sufficient provided that the interactive *FTP* mode (prompting for individual files) has been turned off. In more complex situations it is helpful to write a proper *FTP* script and to invoke it using *FTP take* command or its equivalent.

The *ftp* utility should be invoked while the current directory on the target system is the target directory to receive the data, or the *FTP* script should contain appropriate directory

navigation. Proper file locating can be verified later on by using *Windows File Manager* utility.

4.1.3. CD-R recording

When the complete set of properly located files is present on a buffer disk, the system is ready for recording. The machine needs to be brought to DOS/Windows mode (see section 4.1.1) to accomplish recording.

The first step is to place a disk description contents file *contents.txt* into the root directory of the buffer drive. This file contains a text describing the disk. Sample descriptions may be copied from existing CD's and modified accordingly.

Next, locate a program group *RCD-PC*, and the *RCD-PC* program icon inside. Run the *RCD-PC*; this will open the *File Manager* as well. Go to the *File Manager* window, and inspect the directory structure and its contents. Individual file sizes should exactly match those on on-line archive; wrong file size signals either unnoticed data transmission problem or wrong transmission mode (such as *ascii* transfer for binary data).

Go to the *CD-image/Open* menu item from the *RCD-PC* main menu. Select *ecofs-in.vcd* from a *File Open* dialog box that pops up. This will bring the "image" of a similar CD-R that was recorded the last time. The image contains information on how a CD should be recorded (such as size of a CD/recording time, type of data to be recorded, etc.), and a list of files to be written to the CD. The structure of the former record becomes visible in the *RCD-PC* window and needs to be deleted. Next, new structure needs to be imported by dragging files or directories from *File Manager* to the *RCD-PC* window (you need to delete *contents.txt* from the image and drag a new version in, as the file contents changes). Note that the status line of the *RCD-PC* window contains a size information that changes while files are added to the image. This size should not exceed 620 MB in any case, and some smaller values are safer. Now go to the *ADMI* menu and select *Write*. A dialog box entitled *Source* will appear. Make sure that the *Virtual* radio-button is selected and the *Fix-up* and *Mount* checkboxes are checked, and the number of copies is correct (usually, 1). Click on *OK*, which brings *Global Date Change*. Select *Global Date & Time* radio button, enter date and time of the recording. Click on *OK*. A *Volume Descriptors* dialog box appears with its contents used for previous recording session. Change *Volume Name* to reflect contents of the current disk, you may also enter other necessary changes. Note that the *Abstract File Name* points to the file *contents.txt* created earlier. Clicking on *OK* brings *Write* dialog box. You need to have *Write Data*, *Verify Data*, *Close Session*, *Finalize Disc*, *Progress Meter*, and *Copy Permitted* checkboxes checked, and 1X speed and Mode1 (2048) selected. Two buttons, *Test* and *Write* may be used to progress to the last dialog box; before pressing them, a new CD-R should be placed in a caddy and inserted to the drive. The last dialog box gives you the last chance to withdraw before physical writing starts; click on *Start* to initiate recording. A progress bar will appear,

showing the amount of time to complete the operation. Upon finishing, the disk is ejected from the drive, and the status message appears on the screen.

Even if you checked the *Verify Data*, checkbox which causes verifying the data read from the CD after writing against those remaining on the buffer disk, there still is a possibility that the data were distorted before they arrived to the buffer. Thus, it is necessary to perform a manual verification.

4.1.4. Verifying the disc

Bring a disk to a CD-equipped machine and check if you can read the disk, the description file *contents.txt* , and that you can move across the directory tree, having all the contents of directories listed properly. Next, transfer some of the GRIB files back to the Cray and match them against original data, using the *cmp* compare utility. You should always try to test the file that was written last (usually, the last file in the last directory), as it is most likely to be damaged, e.g. by insufficient space on a CD (Test recording operation was found not to be critical, so there is a possibility of losing some data despite positive result of the test).

4.2. Backing data to tapes

The following script (written by C. Peters), after appropriate modifications, may be used to copy the data to the Cray REEL tapes:

```
#QSUB -q tape -eo -o hist.out -lUb 1
#QSUB -x
#QSUB -s /bin/sh
#QSUB
#
# Use imported variables VOLUME and MONTH to dump meso sfc grib
# files to CART90 tape
#
set -xS
export VOLUME YYMM

mkdir /tmp/wd20cp
cd /tmp/wd20cp
pid=$$

let "icnt=1"
let "num=1"

## Begin tape processing

rsv CART90
vset=$VOLUME

while [ $icnt -le $num ] ; do

if [ $icnt -eq 1 ] ; then
```

```
  file=mesosfc.$YYMM
fi

cd /tmp/wd20cp/$file

tpmnt -X -g CART90 -H 755 -S $vset    \
     -n -u -q n -h 755 -P $file -x 99365
err=$?
echo $err
if [ $err -ne 0 ] ; then
echo " ***** ERROR $err ***** "
kill -9 ${pid}
fi

tar -cvf $file *
##rls -p $file -k

let "icnt = icnt + 1"
done
echo "All done"
rls -a
```

To retrieve, use the following script (written by C. Peters), as a template:

```
# QSUB -lM 1MW -lT 600 -r cfsget
# QSUB -eo -x
# QSUB -s /bin/sh
set -xS

CPDIR=/wd2/wd20/wd20cp/script/tapejobs
export CPDIR VOLUME YYMM
echo $VOLUME
echo $YYMM

mkdir -p /tmp/wd20cp/mesosfc.$YYMM
cd /dm/cfspom/CFS/archive/etasrf/$YYMM
dmget *.grb
cp *.grb /tmp/wd20cp/mesosfc.$YYMM

cd $CPDIR
qsub -x cfs.meso
```

l.

A Description of a Three-Dimensional Coastal Ocean Circulation Model

Alan F. Blumberg and George L. Mellor

Reprinted from:
Three Dimensional Coastal Ocean Models, by Norman S. Heaps (Editor), 1-16.
1987 American Geophysical Union, Washington, DC.

A DESCRIPTION OF A THREE-DIMENSIONAL COASTAL OCEAN CIRCULATION MODEL

Alan F. Blumberg[1]

Dynalysis of Princeton, Princeton, New Jersey 08540

George L. Mellor

Geophysical Fluid Dynamics Program, Princeton University, Princeton, New Jersey 08544

Abstract. A three-dimensional, primitive equation, time-dependent, σ coordinate, free surface, estuarine and coastal ocean circulation model is described in detail. An apparently unique feature is its imbedded turbulent closure submodel which on the basis of previous studies should yield realistic, Ekman surface and bottom layers. The model has been designed to represent ocean physics as realistically as possible given the present-day state of the art and to address phenomena of 1-100 km length and tidal-monthly time scales depending on basin size and grid resolution. The prognostic variables are the three components of the velocity field, temperature, salinity, and two quantities which characterize the turbulence, the turbulence kinetic energy and the turbulence macroscale. The governing equations together with their boundary conditions are solved by finite difference techniques. A horizontally and vertically staggered lattice of grid points is used for the computations. An implicit numerical scheme in the vertical direction and a mode splitting technique in time have been adopted for computational efficiency. The numerics have been designed to readily accommodate the highly time-dependent and often nonlinear processes of coastal upwelling and eddy dynamics. The numerical model incorporates realistic coastline and bottom topography. The actual computer code is configured to take advantage of the array processing design of modern computers so that long-term integrations are possible at tolerable cost. Applications of the model to a variety of coastal settings all produce circulation predictions which seem quite realistic when compared to the available data/theory. These applications include a simulation of the tides in the Chesapeake Bay, a simulation of the coastal circulation off Long Island, New York, and a computation of the general circulation in the Middle and South Atlantic Bights and in the Gulf of Mexico. The grid spacings have ranged from 1 to 50 km in these applications. In a new application to coastal upwelling, the model's behavior is in accord with recently developed ideas of coastal trapped waves.

1. Introduction

The coastal ocean is a region receiving a great deal of attention due to an increasing utilization of its resources. The demands for increasing development have directed both governments and individuals to investigate the basic mechanisms which govern the circulation over the continental shelf. A knowledge of the circulation is useful to the management of fisheries and of oil and gas resource development. Spills of oil and other material from offshore drilling and oil transport activities may occur and significantly affect the environment. Therefore, the movement of these pollutants becomes an important item to predict.

The purpose of this paper is to provide a relatively detailed description of a numerical circulation model developed over the last few years at Princeton University and Dynalysis of Princeton. The model belongs to that class of models where model realism is an important goal and addresses mesoscale phenomena, that is activity characterized by 1-100 km length and tidal-30 day time scales commonly observed in estuaries and the coastal ocean [Beardsley and Boicourt, 1981]. It is envisioned that the model ultimately will be used as part of a coastal ocean forecasting program. The model is a three-dimensional coastal ocean model, incorporating a turbulence closure model to provide a realistic parameterization of the vertical mixing processes. The prognostic variables are the three components of velocity, temperature, salinity, turbulence kinetic energy, and turbulence macroscale. The momentum equations are nonlinear and incorporate a variable Coriolis parameter. Prognostic equations governing the thermodynamic quantities, temperature, and salinity account for water mass variations brought about by highly time-dependent coastal upwelling

---

[1]Now at HydroQual, Inc., Mahwah, NJ 07430.

processes as well as horizontal advective processes. Free surface elevation is also calculated prognostically, with only some sacrifice in computational time so that tides and storm surge events can also be simulated. This is accomplished by use of a mode splitting technique whereby the volume transport and vertical velocity shear are solved separately. Other computer variables include the density, vertical eddy viscosity, and vertical eddy diffusivity. The model also accommodates realistic coastline geometry and bottom topography.

The model performance has been tested in a variety of applications which will not be described. To gain some appreciation for the model's ability to simulate coastal circulation the reader is referred to Blumberg [1977], and Blumberg and Mellor [1979a, b, 1980, 1981a, b, 1983]. These applications include a simulation of the tides in the Chesapeake Bay, a simulation of the coastal circulation off Long Island, New York, and a computation of the general circulation in the Middle Atlantic and South Atlantic Bights and in the Gulf of Mexico. The grid spacings have ranged from 1 to 50 km in these applications. Additional numerical experiments involving upwelling and coastal trapped waves will be described in this paper to provide an illustration of the utility of the model.

2. The Governing Equations

Dynamic and Thermodynamic Equations

The equations which form the basis of the circulation model describe the velocity and surface elevation fields, and the salinity and temperature fields. Two simplifying approximations are used [Bryan, 1969]; first, it is assumed that the weight of the fluid identically balances the pressure (hydrostatic assumption), and second, density differences are neglected unless the differences are multiplied by gravity (Boussinesq approximation).

Consider a system of orthogonal Cartesian coordinates with x increasing eastward, y increasing northward, and z increasing vertically upwards. The free surface is located at $z = \eta(x,y,t)$ and the bottom is at $z = -H(x,y)$. If $\underset{\rightarrow}{V}$ is the horizontal velocity vector with components (U,V) and ∇ the horizontal gradient operator, the continuity equation is

$$\nabla \cdot \underset{\rightarrow}{V} + \frac{\partial W}{\partial z} = 0 \qquad (1)$$

The Reynolds momentum equations are

$$\frac{\partial U}{\partial t} + \underset{\rightarrow}{V} \cdot \nabla U + W \frac{\partial U}{\partial z} - fV = -\frac{1}{\rho_o}\frac{\partial P}{\partial x} + \frac{\partial}{\partial z}\left(K_M \frac{\partial U}{\partial z}\right) + F_x \qquad (2)$$

$$\frac{\partial V}{\partial t} + \underset{\rightarrow}{V} \cdot \nabla V + W \frac{\partial V}{\partial z} + fU = -\frac{1}{\rho_o}\frac{\partial P}{\partial y} + \frac{\partial}{\partial z}\left(K_M \frac{\partial V}{\partial z}\right) + F_y \qquad (3)$$

$$\rho g = -\frac{\partial P}{\partial z} \qquad (4)$$

with ρ_o the reference density, ρ the in situ density, g the gravitational acceleration, P the pressure, K_M the vertical eddy diffusivity of turbulent momentum mixing. A latitudinal variation of the Coriolis parameter, f, is introduced by use of the β plane approximation.

The pressure at depth z can be obtained by integrating the vertical component of the equation of motion, (4), from z to the free surface η, and is

$$P(x,y,z,t) = P_{atm} + g\rho_o\eta + g\int_z^0 \rho(x,y,z',t)dz' \qquad (5)$$

Henceforth, the atmospheric pressure, P_{atm} is assumed constant.

The conservation equations for temperature and salinity may be written as

$$\frac{\partial \theta}{\partial t} + \underset{\rightarrow}{V} \cdot \Delta\theta + W\frac{\partial \theta}{\partial z} = \frac{\partial}{\partial z}\left(K_H \frac{\partial \theta}{\partial z}\right) + F_\theta \qquad (6)$$

$$\frac{\partial S}{\partial t} + \underset{\rightarrow}{V} \cdot \nabla S + W\frac{\partial S}{\partial z} = \frac{\partial}{\partial z}\left(K_H \frac{\partial S}{\partial z}\right) + F_S \qquad (7)$$

where θ is the potential temperature (or in situ temperature for shallow water applications) and S is the salinity. The vertical eddy diffusivity for turbulent mixing of heat and salt is denoted as K_H. Using the temperature and salinity, the density is computed according to an equation of state of the form

$$\rho = \rho(\theta,S) \qquad (8)$$

given by Fofonoff [1962]. The potential density is ρ, that is, the density evaluated as a function of potential temperature and salinity but at atmospheric pressure; it provides accurate density information to calculate horizontal baroclinic gradients which enter in the pressure gradient terms and the vertical stability of the water column which enters into the turbulence closure model even in deep water when pressure effects become important.

All of the motions induced by small-scale processes not directly resolved by the model grid (subgrid scale) is parameterized in terms of horizontal mixing processes. The terms F_x, F_y, F_θ and

found in (2), (3), (6), and (7) represent these resolved processes and in analogy to molecular diffusion can be written as

$$F_x = \frac{\partial}{\partial x}\left[2A_M \frac{\partial U}{\partial x}\right] + \frac{\partial}{\partial y}\left[A_M\left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}\right)\right] \quad (9a)$$

$$F_y = \frac{\partial}{\partial y}\left[2A_M \frac{\partial V}{\partial y}\right] + \frac{\partial}{\partial x}\left[A_M\left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}\right)\right] \quad (9b)$$

d

$$F_{\theta,S} = \frac{\partial}{\partial x} A_H \frac{\partial(\theta,S)}{\partial x} + \frac{\partial}{\partial y} A_H \frac{\partial(\theta,S)}{\partial y} \quad (10)$$

should note that F_x and F_y are invariant to rdinate rotation. While, these horizontal ffusive terms are meant to parameterize subgrid e processes, in practice the horizontal dif- vities, A_M and A_H, are usually required to small-scale computational noise. The form of F_y and $F_{\theta,S}$ allows for variable A_M and A_H but s far they have been held constant. The diffu- ties are chosen so that they do not produce essive smoothing of real features. Values as as $10\ m^2/s$ have been used successfully in ous applications. The relatively fine verti- resolution used in the applications resulted reduced need for horizontal diffusion because izontal advection followed by vertical mixing ectively acts like horizontal diffusion in a l physical sense. An enhancement, now in prog- s, is to relate A_M and A_H to the scales of ion being resolved in the model and to the l deformation field as suggested by Smagorin- [1963].

rbulence Closure

The governing equations contain parameterized ynolds stress and flux terms which account for turbulent diffusion of momentum, heat, and t. The parameterization of turbulence in the el described here is based on the work of Mel- r and Yamada [1974].

The vertical mixing coefficients, K_M and K_H, in , (3), (6), and (7) are obtained by appealing a second order turbulence closure scheme [Mel- r and Yamada, 1982] which characterizes the rbulence by equations for the turbulence kinetic ergy, $q^2/2$, and a turbulence macroscale, according to,

$$\frac{\partial q^2}{\partial t} + \underset{\sim}{V}\cdot\nabla q^2 + W\frac{\partial q^2}{\partial z} = \frac{\partial}{\partial z}\left(K_q \frac{\partial q^2}{\partial z}\right)$$

$$2K_M\left[\left(\frac{\partial U}{\partial z}\right)^2 + \left(\frac{\partial V}{\partial z}\right)^2\right] + \frac{2g}{\rho_o} K_H \frac{\partial \rho}{\partial z} - \frac{2q^3}{B_1 \ell} + F_q \quad (11)$$

d

$$\frac{\partial q^2 \ell}{\partial t} + \underset{\sim}{V}\cdot\nabla(q^2\ell) + W\frac{2q^2\ell}{\partial z}$$

$$= \frac{\partial}{\partial z}\left[K_q \frac{\partial}{\partial z}(q^2\ell)\right] + \ell E_1 K_M\left[\left(\frac{\partial U}{\partial z}\right)^2 + \left(\frac{\partial V}{\partial z}\right)^2\right]$$

$$+ \frac{\ell E_1 g}{\rho_o} K_H \frac{\partial \rho}{\partial z} - \frac{q^3}{B_1}\tilde{W} + F_\ell \quad (12)$$

where a wall proximity function is defined as

$$\tilde{W} \equiv 1 + E_2\left(\frac{\ell}{\kappa L}\right)^2 \quad (13)$$

and where

$$(L)^{-1} \equiv (\eta - z)^{-1} + (H + z)^{-1} \quad (14)$$

Near surfaces it may be shown that both ℓ/κ and L are equal to the distance from the surface ($\kappa = 0.4$ is the von Karman constant) so that $\tilde{W} = 1 + E_2$. Far from the surfaces where $\ell << L$, $W^2 = 1$. The length scale provided by (12) is a characteristic length of the turbulent motion at any point in space or time. An alternative to (12) is to use a transport equation for the dissipation rate [Hanjalic and Launder, 1972]. The former approach according to Mellor and Herring [1973] and Mellor and Yamada [1982] is more consistent since it uses an equation which describes large-scale turbulence to determine the turbulent macroscale. The terms F_q and F_ℓ in (11) and (12) are the horizontal mixing and are parameterized analogously to temperature and salinity by using (10).

While details of the closure model are rather involved, it is possible to reduce the prescription of the mixing coefficients K_M, K_H, and K_q to the following expressions,

$$K_M \equiv \ell q S_M \quad (15a)$$

$$K_H \equiv \ell q S_H \quad (15b)$$

$$K_q \equiv \ell q S_q \quad (15c)$$

The stability functions, S_M, S_H, and S_q are analytically derived, algebraic relations functionally dependent upon $\partial U/\partial z$, $\partial V/\partial z$, $g\rho_o^{-1}\partial\rho/\partial z$, q and ℓ. These relations derive from closure hypotheses described by Mellor [1973] and recently summarized by Mellor and Yamada [1982].

It is convenient to define

$$G_m \equiv \frac{\ell^2}{q^2}\left[\left(\frac{\partial U}{\partial z}\right)^2 + \left(\frac{\partial V}{\partial z}\right)^2\right]^{1/2} \qquad (16a)$$

$$G_H \equiv \frac{\ell^2}{q^2}\frac{g}{\rho_o}\frac{\partial \rho}{\partial z} \qquad (16b)$$

Then the stability functions become

$$S_M\left[6A_1A_2G_M\right] + S_H\left[1 - 2A_2B_2G_H - 12A_1A_2G_H\right] = A_2 \qquad (17a)$$

$$S_M\left[1 + 6A_1^2G_M - 9A_1A_2G_H\right] - S_H\left[12A_1^2G_H + 9A_1A_2G_H\right] = A_1(1 - 3C_1) \qquad (17b)$$

$$S_q = 0.20 \qquad (17c)$$

which are readily solved for S_M and S_H as functions of G_M and G_H. By appealing to laboratory data [Mellor and Yamada, 1982] (see section 6 for further practical details), the empirical constants were assigned the values

$$(A_1, A_2, B_1, B_2, C_1) = (0.92,\ 0.74,\ 16.6,\ 10.1,\ 0.08) \qquad (18)$$

and

$$(E_1, E_2) = (1.8,\ 1.33) \qquad (19)$$

Boundary Conditions

The boundary conditions at the free surface, $z = \eta(x,y)$, are

$$\rho_o K_M\left(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z}\right) = (\tau_{ox}, \tau_{oy}) \qquad (20a)$$

$$\rho_o K_H\left(\frac{\partial \theta}{\partial z}, \frac{\partial S}{\partial z}\right) = (\dot{H}, \dot{S}) \qquad (20b)$$

$$q^2 = B_1^{2/3} u_{\tau s}^2 \ell \qquad (20c)$$

$$q^2\ell = 0 \qquad (20d)$$

$$W = U\frac{\partial \eta}{\partial x} + V\frac{\partial \eta}{\partial y} + \frac{\partial \eta}{\partial t} \qquad (20e)$$

where (τ_{ox}, τ_{oy}) is the surface wind stress vector with the friction velocity, $u_{\tau s}$, the magnitude of the vector. It is doubtful that the mixing length goes to zero at a surface containing wind induced waves as suggested by (20d). The error is incurred in the near surface layers of thickness of order of the wave height. This is an area where further improvement is necessary. The quantity $B_1^{2/3}$ is an empirical constant (6.51) arising from the turbulence closure relations. The net ocean heat flux is $\dot{H}$ and here $\dot{S} \equiv S(0)\,[\dot{E}-\dot{P}]/\rho_o$ where $(\dot{E}-\dot{P})$ is the net evaporation-precipitation fresh water surface mass flux rate and $S(0)$ is the surface salinity. On the side walls and bottom of the basin, the normal gradients of θ and S are zero so that there are no advective and diffusive heat and salt fluxes across these boundaries. At the lower boundary (b),

$$\rho_o K_M\left(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z}\right) = (\tau_{bx}, \tau_{by}) \qquad (21a)$$

$$q^2 = B_1^{2/3} u_{\tau b}^2 \qquad (21b)$$

$$q^2\ell = 0 \qquad (21c)$$

$$W_b = -U_b\frac{\partial H}{\partial x} - V_b\frac{\partial H}{\partial y} \qquad (21d)$$

where $H(x,y)$ is the bottom topography and $u_{\tau b}$ is the friction velocity associated with the bottom frictional stress (τ_{bx}, τ_{by}). The bottom stress is determined by matching velocities with the logarithmic law of the wall. Specifically,

$$\vec{\tau}_b = \rho_o C_D |\vec{V}_b| \vec{V}_b \qquad (22)$$

with value of the drag coefficient C_D given by

$$C_D = \left[\frac{1}{\kappa}\ln(H + z_b)/z_o\right]^{-2} \qquad (23a)$$

where z_b and $\vec{V}_b$ are the grid point and corresponding velocity in the grid point nearest the bottom and κ is the von Karman constant. The final result of (22) and (23) in conjunction with the turbulent closure derived K_M is that the calculations will yield

$$\vec{V} = (\vec{\tau}_b/\kappa u_{\tau b})\ln(z/z_o) \qquad (23b)$$

in the lower boundary region if enough resolution is provided. In those instances where the bottom boundary layer is not well resolved it is more appropriate to specify $C_D = 0.0025$. The actual algorithm is to set C_D to the larger of the two values given by (23a) and 0.0025. The parameter z_o depends on the local bottom roughness; in the absence of specific information $z_o = 1$ cm is used as suggested by Weatherly and Martin [1978].

Open lateral boundary conditions are problematic since one must parameterize the environment

exterior to the relevant domain. Two types of open boundaries exist, inflow and outflow. Temperature and salinity are prescribed from data at an inflowing boundary, whereas at outflow boundaries,

$$\frac{\partial}{\partial t}(\theta,S) + U_n \frac{\partial}{\partial n}(\theta,S) = 0 \qquad (23c)$$

is solved where the subscript n is the coordinate normal to the boundary. Turbulence kinetic energy and the macroscale quantity ($q^2\ell$) are calculated with sufficient accuracy at the boundaries by neglecting the advection in comparison with other terms in their respective equations.

The open lateral velocity boundary conditions in some of the applications are computed by using the available hydrographic data in conjunction with a simplified diagnostic model. This type of model uses only geostrophic plus Ekman dynamics and therefore solves a simplified form of the full equations of motion. It does not require a velocity at a reference level but only along a single transect crossing f/H contours. A detailed description of this model can be found in the work by Kantha et al. [1982]. While the normal component of velocity is specified, a free slip condition is used for the tangential component.

In other applications including those with tidal forcing, either the elevation is prescribed as a function of time and space or a radiation condition of the form

$$\frac{\partial \eta}{\partial t} + c \frac{\partial \eta}{\partial n} = F(s,t) \qquad (24)$$

is prescribed. Here c is the local shallow water wave speed, $(gH)^{1/2}$, and s is the tangential coordinate. The function F(s,t) incorporates the necessary forcing due to tides and the mean calculation as described by Blumberg and Kantha [1985]. The nonlinear terms in the momentum equations are additionally neglected at the open boundary.

Diagnostic Mode

Numerical experiments have been performed [Blumberg and Mellor, 1983] which involve the use of the circulation model in both prognostic and diagnostic modes. In the prognostic mode the momentum equations as well as equations governing the temperature and salinity distributions are integrated as an initial value problem. These predictive experiments do not always reach steady state since the oceanic response time for the density field can be considerable. As an alternative, diagnostic computations are considered.

In the diagnostic mode the observed density distribution is specified at all points in the grid and held fixed in time. The velocity field consistent with this constraint is allowed to spin up from rest. These experiments typically attain steady conditions after 10 days for even fairly large oceanic regions. The diagnostic approach not only provides a powerful tool for deducing the circulation but it also provides a consistent way of initializing a prognostic forecast model.

3. Vertical Coordinate Representation

It has often been noted that the ordinary x,y,z coordinate system has certain disadvantages in the vicinity of large bathymetric irregularities. It is desirable to introduce a new set of independent variables that transforms both the surface and the bottom into coordinate surfaces [Phillips, 1957]. The governing external and internal mode equations are transformed from (x,y,z,t) to (x*,y*,σ,t*) coordinates, where

$$x^* = x \qquad y^* = y \qquad \sigma = \frac{z-\eta}{H+\eta} \qquad t^* = t \qquad (25)$$

Now let $D \equiv H + \eta$ and apply the chain rule; the following relationships linking derivatives in the old system to those in the new system are obtained:

$$\frac{\partial G}{\partial x} = \frac{\partial G}{\partial x^*} - \frac{\partial G}{\partial \sigma}\left(\frac{\sigma}{D}\frac{\partial D}{\partial x^*} + \frac{1}{D}\frac{\partial \eta}{\partial x^*}\right) \qquad (26a)$$

$$\frac{\partial G}{\partial y} = \frac{\partial G}{\partial y^*} - \frac{\partial G}{\partial \sigma}\left(\frac{\sigma}{D}\frac{\partial D}{\partial y^*} + \frac{1}{D}\frac{\partial \eta}{\partial y^*}\right) \qquad (26b)$$

$$\frac{\partial G}{\partial z} = \frac{1}{D}\frac{\partial G}{\partial \sigma} \qquad (26c)$$

$$\frac{\partial G}{\partial t} = \frac{\partial G}{\partial t^*} - \frac{\partial G}{\partial \sigma}\left(\frac{\sigma}{D}\frac{\partial D}{\partial t^*} + \frac{1}{D}\frac{\partial \eta}{\partial t^*}\right) \qquad (26d)$$

where G is an arbitrary field available, and σ ranges from σ = 0 at z = η to σ = −1 at z = −H. A new vertical velocity can now be defined

$$\omega \equiv W - U\left(\sigma\frac{\partial D}{\partial x^*} + \frac{\partial \eta}{\partial x^*}\right) - V\left(\sigma\frac{\partial D}{\partial y^*} + \frac{\partial \eta}{\partial y^*}\right) - \left(\sigma\frac{\partial D}{\partial t^*} + \frac{\partial \eta}{\partial t^*}\right) \qquad (27)$$

which transforms the boundary conditions, (20e) and (21d), into

$$\omega(x^*,y^*,0,t^*) = 0 \qquad (28a)$$

$$\omega(x^*,y^*,-1,t^*) = 0 \qquad (28b)$$

Also, any vertically integrated quantity, G, for example, now appears as

$$\bar{G} = \int_{-1}^{0} G\, d\sigma \qquad (29)$$

Equations (1), (2), (3), (6), (7), (11), and (12) may now be written as (all asterisks will be dropped for notational convenience)

$$\frac{\partial \eta}{\partial t} + \frac{\partial UD}{\partial x} + \frac{\partial VD}{\partial y} + \frac{\partial \omega}{\partial \sigma} = 0 \tag{30}$$

$$\frac{\partial UD}{\partial t} + \frac{\partial U^2 D}{\partial x} + \frac{\partial UVD}{\partial y} + \frac{\partial U\omega}{\partial \sigma} - fVD + gD\frac{\partial \eta}{\partial x} = \frac{\partial}{\partial \sigma}\left[\frac{K_M}{D}\frac{\partial U}{\partial \sigma}\right] - \frac{gD^2}{\rho_o}\frac{\partial}{\partial x}\int_\sigma^o \rho\, d\sigma + \frac{gD}{\rho_o}\frac{\partial D}{\partial x}\int_\sigma^o \sigma\frac{\partial \rho}{\partial \sigma}\, d\sigma + DF_x \tag{31}$$

$$\frac{\partial VD}{\partial t} + \frac{\partial UVD}{\partial x} + \frac{\partial V^2 D}{\partial y} + \frac{\partial V\omega}{\partial \sigma} + fUD + gD\frac{\partial \eta}{\partial y} = \frac{\partial}{\partial \sigma}\left[\frac{K_M}{D}\frac{\partial V}{\partial \sigma}\right] - \frac{gD^2}{\rho}\frac{\partial}{\partial y}\int_\sigma^o \rho\, d\sigma + \frac{gD}{\rho_o}\frac{\partial D}{\partial y}\int_\sigma^o \sigma\frac{\partial \rho}{\partial \sigma}\, d\sigma + DF_y \tag{32}$$

$$\frac{\partial \theta D}{\partial t} + \frac{\partial \theta UD}{\partial x} + \frac{\partial \theta VD}{\partial y} + \frac{\partial \theta \omega}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left[\frac{K_H}{D}\frac{\partial \theta}{\partial \sigma}\right] + DF_\theta \tag{33}$$

$$\frac{\partial SD}{\partial t} + \frac{\partial SUD}{\partial x} + \frac{\partial SVD}{\partial y} + \frac{\partial S\omega}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left[\frac{K_H}{D}\frac{\partial S}{\partial \sigma}\right] + DF_S \tag{34}$$

$$\frac{\partial q^2 D}{\partial t} + \frac{\partial Uq^2 D}{\partial x} + \frac{\partial Vq^2 D}{\partial y} + \frac{\partial \omega q^2}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left(\frac{K_q}{D}\frac{\partial q^2}{\partial \sigma}\right) + \frac{2K_M}{D}\left[\left(\frac{\partial U}{\partial \sigma}\right)^2 + \left(\frac{\partial V}{\partial \sigma}\right)^2\right] + \frac{2g}{\rho_o}K_H\frac{\partial \rho}{\partial \sigma} - \frac{2Dq^3}{\Lambda_1} + DF_q \tag{35}$$

$$\frac{\partial q^2 \ell D}{\partial t} + \frac{\partial Uq^2 \ell D}{\partial x} + \frac{\partial Vq^2 \ell D}{\partial y} + \frac{\partial \omega q^2}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left(\frac{K_q}{D}\frac{\partial q^2 \ell}{\partial \sigma}\right) + E_1 \ell\left\{\frac{K_M}{D}\left[\left(\frac{\partial U}{\partial \sigma}\right)^2 + \left(\frac{\partial V}{\partial \sigma}\right)^2\right] + \frac{qE_3}{\rho_o}K_H\frac{\partial \rho}{\partial \sigma}\right\} - \frac{Dq^3}{B_1}\tilde{W} + DF_\ell \tag{36}$$

The horizontal viscosity and diffusion terms are defined according to:

$$DF_x \equiv \frac{\partial \hat{\tau}_{xx}}{\partial x} - \frac{\partial}{\partial \sigma}\left[\left(\frac{\sigma}{D}\frac{\partial D}{\partial x} + \frac{1}{D}\frac{\partial \eta}{\partial x}\right)\hat{\tau}_{xx}\right] + \frac{\partial}{\partial y}(\hat{\tau}_{yx}) - \frac{\partial}{\partial \sigma}\left[\left(\frac{\sigma}{D}\frac{\partial D}{\partial y} + \frac{1}{D}\frac{\partial \eta}{\partial y}\right)\hat{\tau}_{yx}\right] \tag{37}$$

$$DF_y \equiv \frac{\partial \hat{\tau}_{yy}}{\partial y} - \frac{\partial}{\partial \sigma}\left[\left(\frac{\sigma}{D}\frac{\partial D}{\partial y} + \frac{1}{D}\frac{\partial \eta}{\partial y}\right)\hat{\tau}_{yy}\right] + \frac{\partial}{\partial x}(\hat{\tau}_{xy}) - \frac{\partial}{\partial \sigma}\left[\left(\frac{\sigma}{D}\frac{\partial D}{\partial x} + \frac{1}{D}\frac{\partial \eta}{\partial x}\right)\hat{\tau}_{xy}\right] \tag{38}$$

with

$$\hat{\tau}_{xx} = 2A_M\left[\frac{\partial UD}{\partial x} - \frac{\partial}{\partial \sigma}\left(\sigma\frac{\partial D}{\partial x} + \frac{\partial \eta}{\partial x}\right)U\right] \tag{39}$$

$$\hat{\tau}_{xy} = \hat{\tau}_{yx} = A_M\left[\frac{\partial UD}{\partial y} - \frac{\partial}{\partial \sigma}\left(\sigma\frac{\partial D}{\partial y} + \frac{\partial \eta}{\partial y}\right)U + \frac{\partial VD}{\partial x} - \frac{\partial}{\partial \sigma}\left(\sigma\frac{\partial D}{\partial x} + \frac{\partial \eta}{\partial x}\right)V\right] \tag{40}$$

$$\hat{\tau}_{yy} = 2A_M\left[\frac{\partial VD}{\partial y} - \frac{\partial}{\partial \sigma}\left(\sigma\frac{\partial D}{\partial y} + \frac{\partial \eta}{\partial y}\right)V\right] \tag{41}$$

Also,

$$DF_{\theta_i} = \frac{\partial \hat{q}_x}{\partial x} - \frac{\partial}{\partial \sigma}\left[\left(\frac{\sigma}{D}\frac{\partial D}{\partial x} + \frac{1}{D}\frac{\partial \eta}{\partial x}\right)\hat{q}_x\right] + \frac{\partial \hat{q}_y}{\partial y} - \frac{\partial}{\partial \sigma}\left[\left(\frac{\sigma}{D}\frac{\partial D}{\partial y} + \frac{\partial \eta}{\partial y}\right)\hat{q}_y\right] \tag{42}$$

$$\hat{q}_x = A_H\left[\frac{\partial \theta_i D}{\partial x} - \frac{\partial}{\partial \sigma}\left(\sigma\frac{\partial D}{\partial x} + \frac{\partial \eta}{\partial x}\right)\theta_i\right] \tag{43}$$

$$\hat{q}_y = A_H\left[\frac{\partial \theta_i D}{\partial y} - \frac{\partial}{\partial \sigma}\left(\sigma\frac{\partial D}{\partial y} + \frac{\partial \eta}{\partial y}\right)\theta_i\right] \tag{44}$$

where θ_i now represents θ, S, q^2 and $q^2\ell$.

Since the time this paper was accepted for publication, Mellor and Blumberg [1985] have shown that the conventional model for horizontal diffusion is incorrect when bottom topographical slopes are large. A new formulation has been suggested which is simpler than equations (37)-(44) and makes it possible to model realistically bottom boundary layers over sharply sloping bottoms.

4. Mode Splitting Technique

The equations governing the dynamics of coastal circulation contain propagation of fast moving external gravity waves and slow moving internal gravity waves. It is desirable in terms of computer economy to separate out vertically integrated equations (external mode) from the vertical structure equations (internal mode). This technique, known as mode splitting [see Simons, 1974; Madala and Piacsek, 1977] permits the calculation of the free surface elevation with little sacrifice in computational time by solving the volume transport separately from the vertical velocity shear.

The volume transport, external mode equations

are obtained by integrating the internal mode equations over the depth, thereby eliminating all vertical structure. By integrating (30) from $\sigma = -1$ to $\sigma = 0$ and using the boundary conditions (28a,b) an equation for the surface elevation can be written as

$$\frac{\partial \eta}{\partial t} + \frac{\partial \bar{U}D}{\partial x} + \frac{\partial \bar{V}D}{\partial y} = 0 \qquad (45)$$

and the momentum equations become upon vertical integration

$$\frac{\partial \bar{U}D}{\partial t} + \frac{\partial \bar{U}^2 D}{\partial x} + \frac{\partial \overline{UV}D}{\partial y} - f\bar{V}D + gD\frac{\partial \eta}{\partial x} - D\bar{F}_x = -\overline{wu}(0)$$

$$+ \overline{wu}(-1) - \frac{\partial \overline{DU'^2}}{\partial x} - \frac{\partial \overline{DU'V'}}{\partial y} - \frac{gD^2}{\rho_o}\frac{\partial}{\partial x}\int_{-1}^{0}\int_{\sigma}^{0} \rho d\sigma' d\sigma$$

$$+ \frac{gD}{\rho_o}\frac{\partial D}{\partial x}\int_{-1}^{0}\int_{\sigma}^{0} \sigma' \frac{\partial \rho}{\partial \sigma} d\sigma' d\sigma \qquad (46)$$

$$\frac{\partial \bar{V}D}{\partial t} + \frac{\partial \overline{UV}D}{\partial x} + \frac{\partial \bar{V}^2 D}{\partial y} + f\bar{U}D + gD\frac{\partial \eta}{\partial y} - D\bar{F}_y = -\overline{wv}(0)$$

$$+ \overline{wv}(-1) - \frac{\partial \overline{DU'V'}}{\partial x} - \frac{\partial \overline{DV'^2}}{\partial y} - \frac{gD^2}{\rho_o}\frac{\partial}{\partial y}\int_{-1}^{0}\int_{\sigma}^{0} \rho d\sigma' d\sigma$$

$$+ \frac{gD}{\rho_o}\frac{\partial D}{\partial y}\int_{-1}^{0}\int_{\sigma}^{0} \sigma' \frac{\partial \rho}{\partial \sigma'} d\sigma' d\sigma \qquad (47)$$

where the pressure has been obtained from (5) and the vertically integrated velocities are defined as

$$(\bar{U},\bar{V}) \equiv \int_{-1}^{0} (U,V)\, d\sigma \qquad (48)$$

The wind stress components are $-\overline{wu}(0)$, and $-\overline{wv}(0)$, and the bottom stress components are $-\overline{wu}(-1)$ and $-\overline{wv}(-1)$. The terms in (46) and (47) involving U'^2, $U'V'$, and V'^2 represent vertical averages of the cross-products of the velocity departures from the vertically integrated (average) velocity and are often denoted as the dispersion terms. Thus

$$(\overline{U'^2}, \overline{V'^2}, \overline{U'V'}) = \int_{-1}^{0} (U'^2, V'^2, U'V')d\sigma \qquad (49)$$

where $(U',V') = (U-\bar{U}, V-\bar{V})$. The quantities $\bar{F}_x$ and $\bar{F}_y$ are vertical integrals of the horizontal momentum diffusion and are defined according to

$$D\bar{F}_x = \frac{\partial}{\partial x}\left(2A_M \frac{\partial \bar{U}D}{\partial x}\right) + \frac{\partial}{\partial y} A_M \left(\frac{\partial \bar{U}D}{\partial y} + \frac{\partial \bar{V}D}{\partial x}\right) \qquad (50)$$

$$D\bar{F}_y = \frac{\partial}{\partial y}\left(2A_M \frac{\partial \bar{V}D}{\partial y}\right) + \frac{\partial}{\partial x} A_M \left(\frac{\partial \bar{U}D}{\partial y} + \frac{\partial \bar{V}D}{\partial x}\right) \qquad (51)$$

The computational strategy is to solve equations for the external mode, the shallow water wave equations (45), (46), and (47), with a short time step to resolve tidal motions. The external mode solutions are obtained with the terms on the right-hand side of (46) and (47) held fixed in time and after a large number of time steps, of the order of 100, an internal mode calculation is carried out. The external mode provides $\partial\eta/\partial x$ and $\partial\eta/\partial y$ for insertion into the internal mode equations, (30) through (36), which are then solved with a much longer time step. Once the vertical structure has been determined, the terms on the right-hand side (46) and (47) are updated and another external mode solution begins. In future simulations, the advective and diffusive terms in (46) and (47) will be supplied by the internal mode.

The external mode equations have not been subtracted from the original equations (30) and (32) to form the more conventional internal mode set as, for example, in Bryan [1969] and Wang [1982]. Consequently there may be a slow tendency for the vertical integral of the internal mode velocities to differ from the external mode velocities. This arises because of different truncation errors in each mode. To insure against accumulated mismatch, the vertical mean of the internal velocity is replaced at every time step by the external mode velocity.

5. Finite Difference Formulation

The governing equations form a set of simultaneous partial differential equations which cannot be solved using known analytic methods. The equations require numeric computational methods using discretized equations on a grid. In anticipation of constructing the finite differencing scheme, the governing equations have been cast into their flux form. This is to insure that certain integral constraints are maintained by the differencing.

Spatial and Temporal Finite Differencing

To derive the finite difference equations, the following sum and difference operations are defined:

$$\overline{F(x,y,\sigma,t)}^x \equiv \frac{F(x+\frac{\Delta x}{2},y,\sigma,t) + F(x-\frac{\Delta x}{2},y,\sigma,t)}{2} \qquad (52a)$$

$$\delta_x F(x,y,\sigma,t) \equiv \frac{F(x+\frac{\Delta x}{2},y,\sigma,t) - F(x-\frac{\Delta x}{2},y,\sigma,t)}{\Delta x} \qquad (52b)$$

$$\delta_x \overline{F(x,y,\sigma,t)}^x \equiv \frac{F(x+\Delta x,y,\sigma,t) - F(x-\Delta x,y,\sigma,t)}{2\Delta x} \qquad (52c)$$

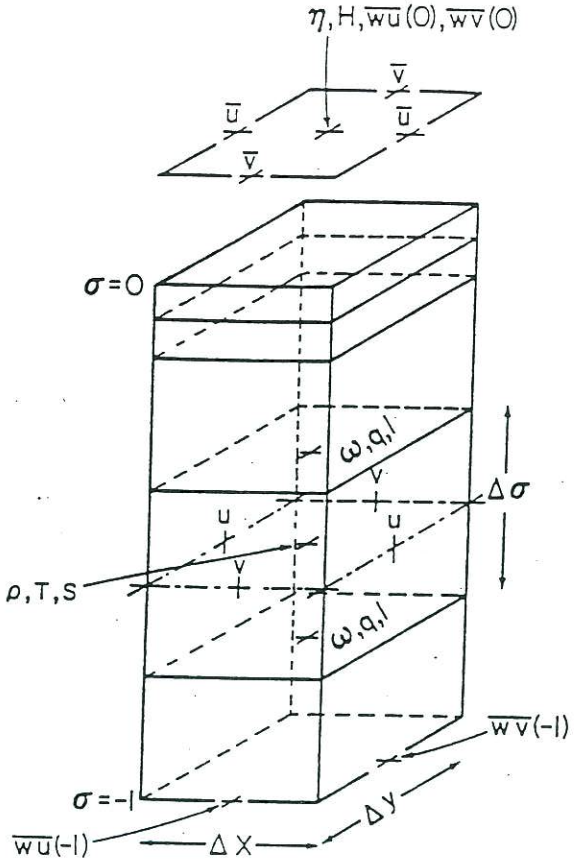


Fig. 1. The locations of the variables on the finite difference grid.

and

$$\overline{F(x,y,\sigma,t)}^{xy} \equiv \overline{\overline{F(x,y,\sigma,t)}^{x}}^{y} \equiv \overline{\overline{F(x,y,\sigma,t)}^{y}}^{x} \qquad (53)$$

the bar and delta (δ) operators form a commutative and distributive algebra. A variable $F(x,y,\sigma,t)$ can now be written as $F^{n}_{i,j,k}$. The relative positions of the variables on the staggered computational "C" grid are shown in Figure 1. The staggered arrangement uses U at points to the east and west of the point where η and H are defined and V at points to the north and south of the η and H points. This type of grid has been shown by Batteen and Han [1981] to be the most effective grid for high resolution (< 50 km grids) ocean circulation models. The Δx and Δy are the constant horizontal grid spacings and $\Delta\sigma$ is the vertical increment which varies in thickness to accommodate more resolution near the surface and bottom.

The finite difference equations governing the motion of the baroclinic (internal) modes, (30), (31), and (32) are

$$\delta_t \eta + \delta_x(\bar{D}^x U) + \delta_y(\bar{D}^y V) + \delta_\sigma(\omega) = 0 \qquad (54)$$

$$\delta_t\overline{(\bar{D}^x U)}^{t} + \delta_x\overline{(\bar{D}^x U\,\bar{U}^x)}^{x} + \delta_y\overline{(\bar{D}^y V\,\bar{U}^y)}^{x} - f\overline{\bar{V}^y D}^{x} + \delta_\sigma(\bar{W}^x\bar{U}^\sigma) + g\bar{D}^x\delta_x\eta = \delta_\sigma\left[\frac{\overline{K_M}^{x}}{\bar{D}^x}\delta_\sigma(U)^{n+1}\right] - \frac{g(\bar{D}^x)^2}{\rho_o}\delta_x\left[\sum_{m=1}^{k}\overline{\rho_{m-1/2}}^{\sigma}\,\overline{\Delta\sigma_{m-1/2}}^{\sigma}\right] + F_x^{n-1} \qquad (55)$$

$$\delta_t\overline{(\bar{D}^y V)}^{t} + \delta_x\overline{(\bar{D}^x U\,\bar{V}^x)}^{y} + \delta_y\overline{(\bar{D}^y V\,\bar{V}^y)}^{y} + \overline{f\bar{U}^x D}^{y} + \delta_\sigma(\bar{W}^y\bar{V}^\sigma) + g\bar{D}^y\delta_y\eta = \delta_\sigma\left[\frac{\overline{K_M}^{y}}{\bar{D}^y}\delta_\sigma(V)^{n+1}\right] - \frac{g(\bar{D}^y)^2}{\rho_o}\delta_y\left[\sum_{m-1}^{k}\overline{\rho_{m-1/2}}^{\sigma}\,\overline{\Delta\sigma_{m-1/2}}^{\sigma}\right] + F_y^{n=1} \qquad (56)$$

The parameter k in (55) and (56) is the number of vertical grid points over which the summation is performed and $\Delta\sigma$ is the spacing of the vertical layers. The superscripts n+1 and n-1 are used to indicate the appropriate time level. All other terms are understood to be a level n. These difference equations are similar to those proposed by Lilly [1965], Leendertse et al. [1973], Holland and Lin [1975], and Blumberg [1977]. To reduce the numerical truncation associated with density gradients in regions of large baroclinic and topographic variability, a reduced density (area mean removed) is introduced to (55) and (56).

The conservation equation for a scalar, θ_i as in (33) and (34) is differenced as

$$\delta_t\overline{(\theta_i D)}^{t} + \delta_x(\bar{\theta}_i^x U\bar{D}^x) + \delta_y(\bar{\theta}_i^y V\bar{D}^y) + \delta_\sigma(\bar{\theta}_i^\sigma\omega) = \delta_\sigma\left(\frac{K_H}{D}\delta_\sigma\theta_i\right)^{n+1} + F_{\theta_i}^{n-1} \qquad (57)$$

The advective characteristics of this particular differencing scheme, evaluated by Kerr and Blumberg [1979], can produce a nonphysical behavior if discontinuities in the property, θ_i, exist. The scheme introduces no artificial horizontal (or vertical) diffusion so that small scale noise generated at a discontinuity must be controlled with the explicit diffusion as is preferred. Similar differencing for the turbulence equations (35) to (36) results for the kinetic energy

$$\overline{\delta_t(q^2 D)}^{t} + \delta_x\overline{(\bar{U}^\sigma q^{2\bar{x}} D^x)}^{x} + \delta_y\overline{(\bar{V}^\sigma q^{2\bar{y}} D^y)}^{x} + \delta_\sigma\overline{\omega^\sigma q^{2\sigma}}^{x} = \delta_\sigma\left(\frac{K_q}{D}\delta_\sigma q^2\right)^{n+1} + \frac{2K_M}{D}\left[(\delta_\sigma\bar{U}^x)^2 + (\delta_\sigma\bar{V}^y)^2\right] + \frac{2g}{\rho_o}K_H\,\delta_\sigma\rho - \frac{2Dq^3}{B_1\ell} + F_q^{n-1} \qquad (58)$$

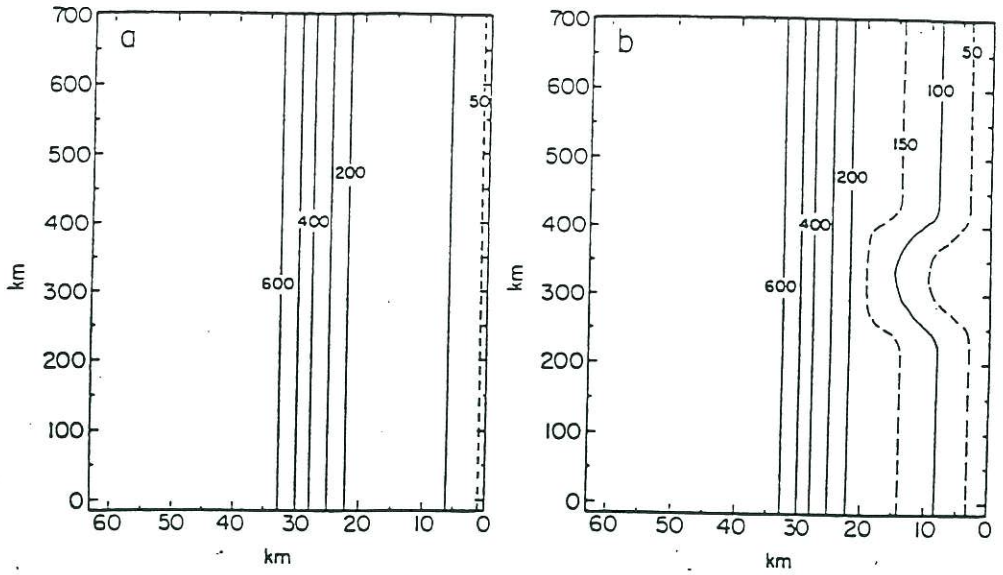


Fig. 2. The bottom topography used in the upwelling experiments. Isobaths in m. (a) The case with the northward invariant topography, and (b) the case with a topographic irregularity.

and for the quantity $q^2 \ell$,

$$\delta_t(\overline{q^2\ell D})^t + \delta_x(\overline{U}^\sigma \overline{q^2\ell}^x \overline{D}^x) + \delta_y(\overline{V}^\sigma \overline{q^2\ell}^y \overline{D}^y) + \delta_\sigma \overline{\omega}^\sigma \overline{q^2\ell}^\sigma$$
$$= \delta_\sigma\left(\frac{K_q}{D}\delta_\sigma q^2\ell\right)^{n+1} + \ell E_1 \frac{K_M}{D}\left[(\delta_\sigma \overline{U}^x)^2 (\delta_\sigma \overline{V}^y)^2\right]$$
$$+ \frac{\ell E_1 g}{\rho_o} K_H \delta_\sigma \rho - \frac{Dq^3}{B_1}\left\{1 + E_2\left[\frac{\ell}{\kappa D}\left(\frac{1}{\sigma} + \frac{1}{1+\sigma}\right)\right]^2\right\} + F_\ell^{n-1} \qquad (59)$$

The implicit treatment of the vertical diffusion terms is used to accommodate the small vertical spacing required to resolve the important top and bottom boundary layers without drastically reducing the time step as would be the case with the more usual explicit schemes. The use of an implicit scheme results in a tri-diagonal matrix which is solved by a Gaussian elimination method [Richtmyer and Morton, 1967].

The external mode equations (45) through (47) are entirely explicit and are differenced for the continuity equation,

$$\delta_t \bar{\eta}^t + \delta_x(\bar{D}^x \bar{U}) + \delta_y(\bar{D}^y V) = 0 \qquad (60)$$

and for the momentum equations

$$\delta_t(\overline{\bar{D}^x \bar{U}})^t + \delta_x(\overline{\bar{D}^x \bar{U}}^{x-x}\bar{U}) + \delta_y(\overline{\bar{D}^y \bar{V}}^{x-y}\bar{U}) - f\overline{\bar{V}^y D}^x$$
$$+ g\bar{D}^x \delta_x \eta - \bar{F}_x^{n-1} = \Phi_x \qquad (61)$$

$$\delta_t(\overline{\bar{D}^y \bar{V}})^t + \delta_x(\overline{\bar{D}^x \bar{U}}^{y-x}\bar{V}) + \delta_y(\overline{\bar{D}^y \bar{V}}^{y-y}\bar{V}) + f\overline{\bar{U} D}^y$$
$$+ g\bar{D}^y \delta_y \eta - \bar{F}_y^{n-1} = \Phi_y \qquad (62)$$

The structure functions, Φ_x and Φ_y, are composed of quantities in (46) and (47) provided by the internal mode and are computed by vertical integration of the corresponding terms in the internal mode equation using the rectangular rule. The finite difference version of the horizontal viscosity and diffusion is not presented here as these terms are exceedingly cumbersome but straightforward.

The finite difference equations presented above can be demonstrated to be of second order accuracy in space and time and to conserve energy, temperature, salinity, mass, and momentum. Finally, the model's computer code has been deliberately designed to be economical on modern array processing computers.

Stability Constraints

The leap-frog differencing used for the time stepping introduces a tendency for the solution at even and odd time steps to split. This time splitting is removed by a weak filter [Asselin, 1972] where the solution is smoothed at each time step according to

$$F_s^n = F^n + \frac{\alpha}{2}(F^{n+1} - 2F^n + F_s^{n-1}) \qquad (63)$$

where $\alpha = 0.05$ and F_s is a smoothed solution.

TABLE 1. The Vertical σ Coordinate Distribution

Level	σ	σ'	Δσ
1	0.00000		
		-0.01042	0.02083
2	-0.02083		
		-0.02946	0.02083
3	-0.04167		
		-0.05893	0.04167
4	0.08333		
		-0.11785	0.08333
5	-0.16667		
		-0.20833	0.08333
6	-0.25000		
		0.29167	0.08333
7	-0.33333		
		-0.37500	0.08333
8	-0.41667		
		-0.45833	0.08333
9	-0.50000		
		-0.54167	0.08333
10	-0.58333		
		-0.62500	0.08333
11	-0.66667		
		-0.70833	0.08333
12	-0.75000		
		-0.79167	0.08333
13	-0.83333		
		-0.88215	0.08333
14	-0.91667		
		-0.94107	0.04167
15	-0.95833		
		-0.97917	0.04167
16	-1.00000		

The quantity σ refers to the depths at which the turbulence quantities and the vertical velocity are located, while σ' corresponds to the depth at which horizontal velocity, temperature, salinity, and denisty are defined. The Δσ is the grid spacing. The relative position of the variables is shown in Figure 1.

This technique introduces less damping than either the Euler-backward or forward stepping techniques.

The Courant-Friedrichs-Levy (CFL) computational stability condition on the vertically integrated, external mode, transport equations limits the time step as shown by Blumberg and Mellor [1981a] according to

$$\Delta t < \frac{1}{C_t}\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)^{-1/2} \qquad (64a)$$

where

$$C_t = 2(gH)^{1/2} + \bar{U}_{max} \qquad (64b)$$

$\bar{U}_{max}$ is the maximum average velocity expected. There are other restrictions but in practice the CFL limit is the most stringent. The model time step is usually 90% of this limit. The internal mode has a much less stringent time step since the fast moving external mode effects have been removed. The time step criteria is analogous to the one for the external mode given by (64a) and is

$$\Delta T < \frac{1}{C_t}\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)^{-1/2} \qquad (65)$$

where $C_T = 2C + U_{max}$, with C being the maximum internal gravity wave speed commonly of order 2m/s and U_{max} is the maximum advective speed. For typical coastal ocean conditions the ratio of the time steps, $\Delta T/\Delta t$, is often a factor of 80-100.

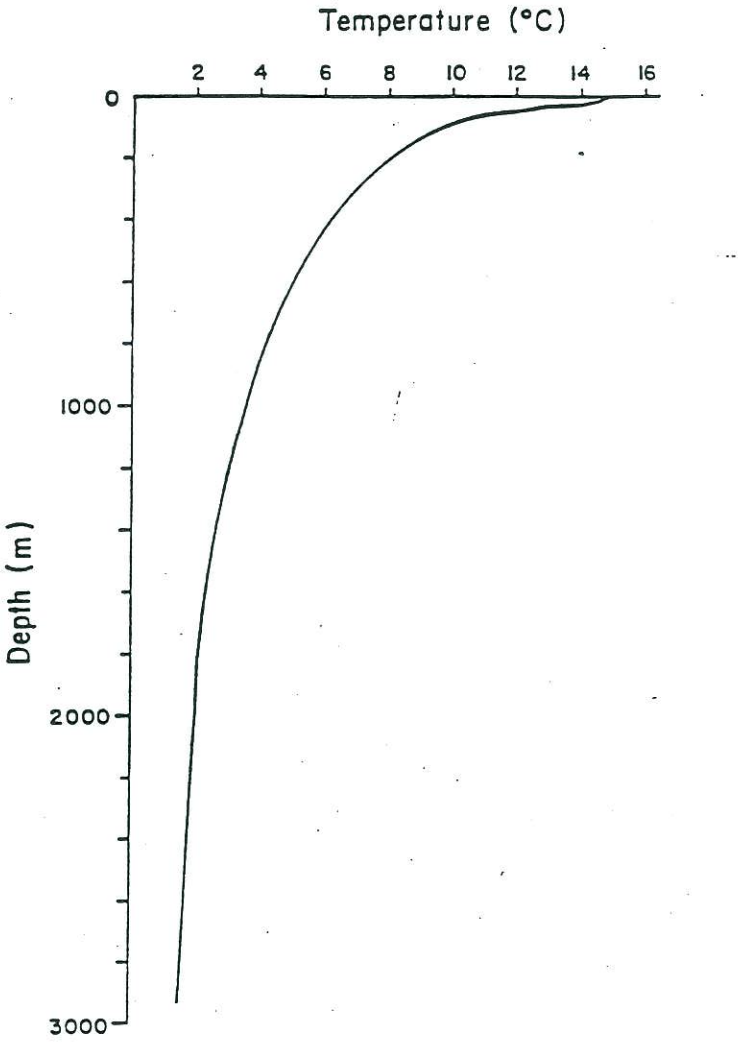


Fig. 3. The initial temperature distribution used in the prognostic model experiments. The distribution is typical of that observed off the coast of California.

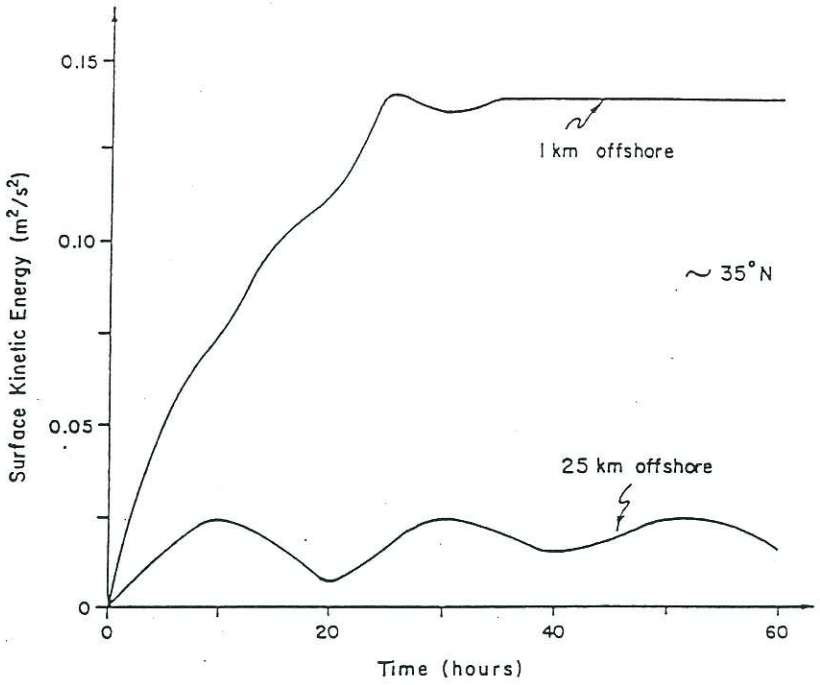


Fig. 4. Time history of the surface kinetic energy at two positions offshore; results from the prognostic model upwelling experiment.

Diffusion is important in the internal mode but does not affect the overall choice of time step, unless the grid Reynolds number is of order 1, in which case

$$\Delta T < \frac{1}{4A_H}\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}\right)^{-1} \qquad (66)$$

must be used.

A rotational condition is

$$\Delta t < \frac{1}{f} = \frac{1}{2\Omega \sin\theta} \qquad (67)$$

where Ω is the angular velocity of the earth and θ is the latitude. However, even for high latitudes (67) is not a limiting factor.

6. Application: Upwelling Coastal Trapped Waves

To illustrate an application of the circulation model, a calculation not previously presented in the literature will be described. For additional applications the reader is referred to the references cited in the introduction. For the purposes of this paper, it is appropriate to investigate in a simpler context some of the characteristics of upwelling and eastern boundary currents. This approach helps to establish model credibility and also provides a framework for extending the model to include geographical settings where the circulation can be indeed complex.

Consider the response of a stratified ocean adjacent to a meridional boundary to the onset of equatorward (upwelling favorable) winds. A rectangular ocean basin 65 km wide and 700 km in north-south extent centered about latitude 36°N is used. The basin has a continental shelf-slope with characteristics typical of northern California as shown in Figure 2a. Alongshore variation in topography is not considered initially. The east-west grid spacing is 2 km and the north-south spacing is 30 km. Because this is a limited domain in spatial extent, that is, an enclosed basin, only short duration experiments can be considered. Otherwise disturbances generated at the boundaries become an important although artificial contribution to the response.

The model consists of 16 vertical levels with irregular vertical spacing in σ space as shown in Table 1. The initial temperature distribution is horizontally homogeneous with a vertical structure similar to that observed off California (~35°N, 125°W). A strong thermocline is present at a depth of ~150 m as shown in Figure 3. The salinity distribution, on the other hand, is set everywhere equal to 34 per mil and used as a check on the conservation properties of the finite difference technique. The horizontal mixing coefficients are both chosen as 50 m^2/s. The external and internal mode time steps are limited to 10 s and 10 min respectively. A meridional, equatorward, 2 $dyne/cm^2$ wind stress is impulsively imposed at the surface in a 240-km-wide zonal band about 200 km from the southern edge of the domain. There is no wind stress curl or heat flux

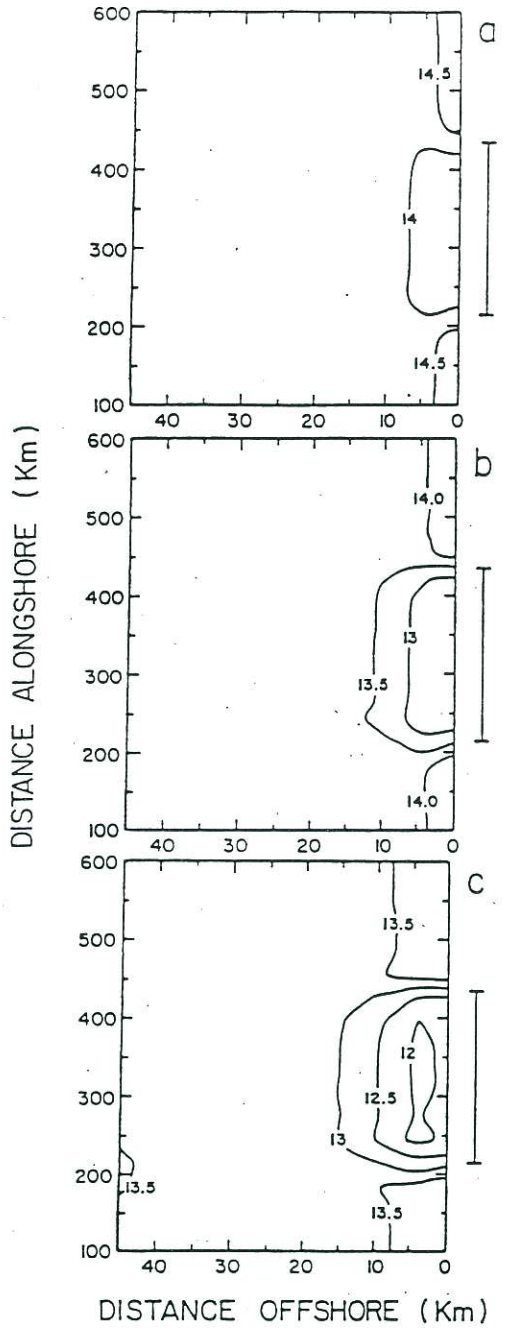


Fig. 5. The surface temperature (°C) distributions from the upwelling experiment after (a) 20 hours, (b) 40 hours, and (c) 60 hours. The forcing region is marked on the right side of each figure.

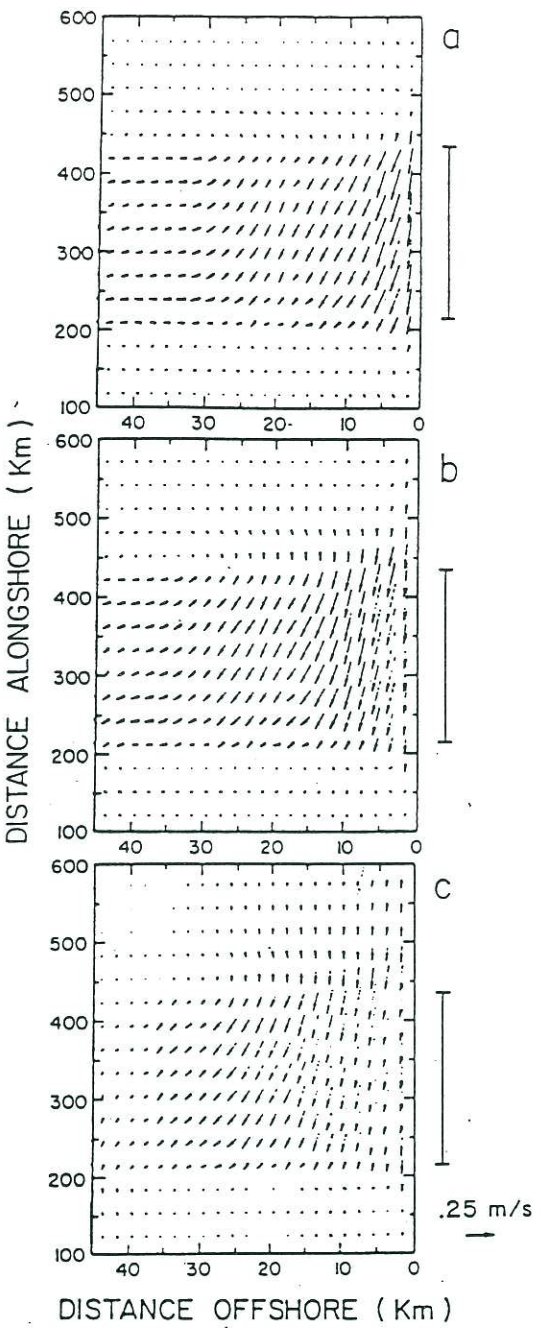


Fig. 6. The near surface circulation patterns from the upwelling experiment after (a) 20 hours, (b) 40 hours, and (c) 60 hours. The forcing region is marked on the right side of each figure.

imposed. The radius of deformation for the initial stratification is ~10 km and is smaller than the width of the shelf-slope, the ratio is ~0.5.

The time history of surface kinetic energy for the 2-1/2-day simulation at locations 1 and 25 km offshore is shown in Figure 4. Inertial motions are clearly evident with the proper 20-hour period (~ 35°N). The shelf water response to the wind is illustrated in Figures 5-9. Sequential patterns at 20-hour intervals spanning 60 hours of the horizontal distributions of surface temperature, surface velocity, and 100-m-depth velocity are depicted. Also included are patterns

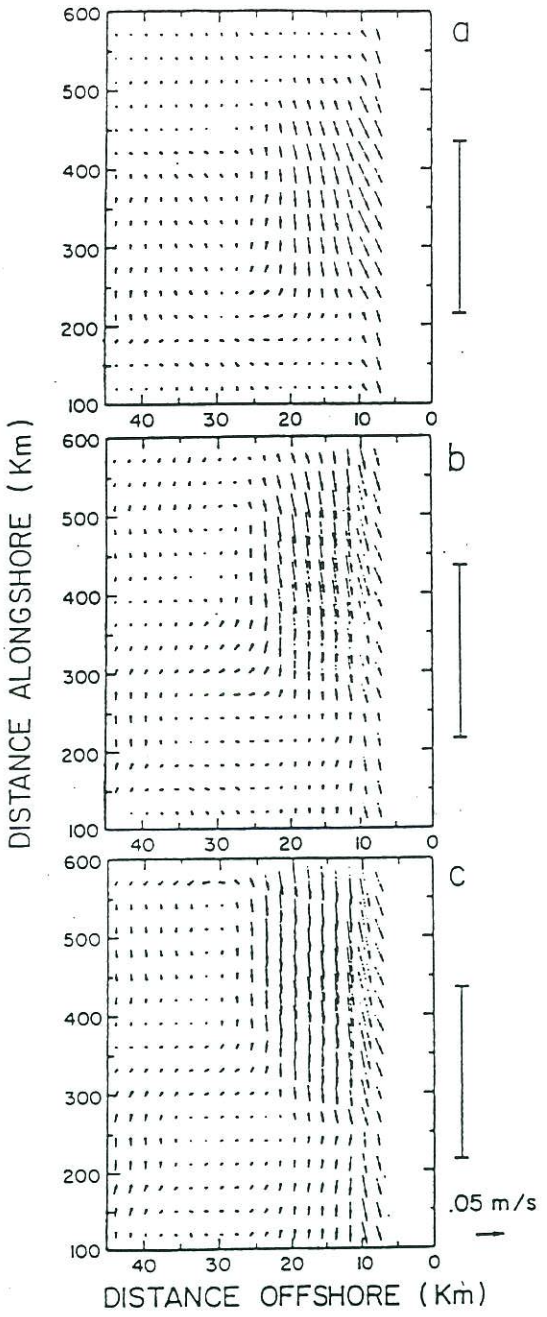


Fig. 7. The 100-m-depth circulation patterns from the upwelling experiment after (a) 20 hours, (b) 40 hours, and (c) 60 hours. The forcing region is marked on the right side of each figure.

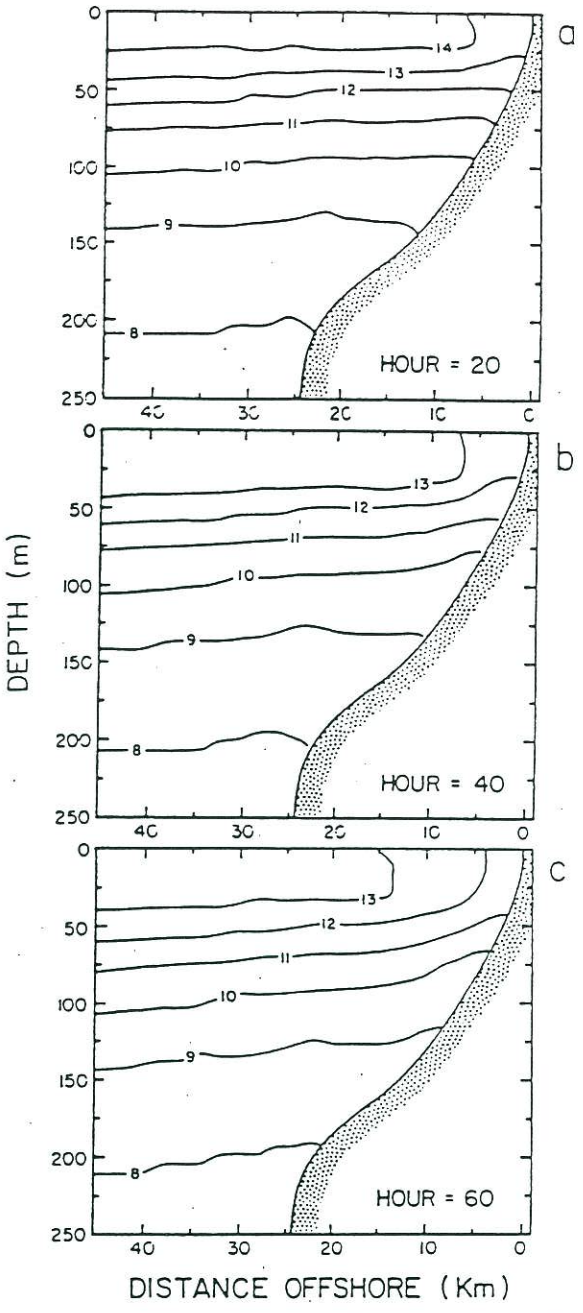


Fig. 8. Sequential patterns of temperature along a vertical section near the northern edge of the wind forcing zone (CI = 1°C).

of the temperature and alongshore velocity from a vertical section near the northern edge of the forcing zone. An intense downwelling response exists on the western boundary. To focus upon the upwelling regime, only the eastern boundary region is shown in these figures.

The initial response to the onset of the winds is the classical Ekman surface offshore flow and the compensating onshore flow at depth. Intense coastal upwelling is found within the forcing zone (see Figures 5 and 8). As time progresses the onshore circulation decreases in strength and the flow becomes markedly three-dimensional. The equatorward jet is confined to the coastal region. As proposed by Philander and Yoon [1982],

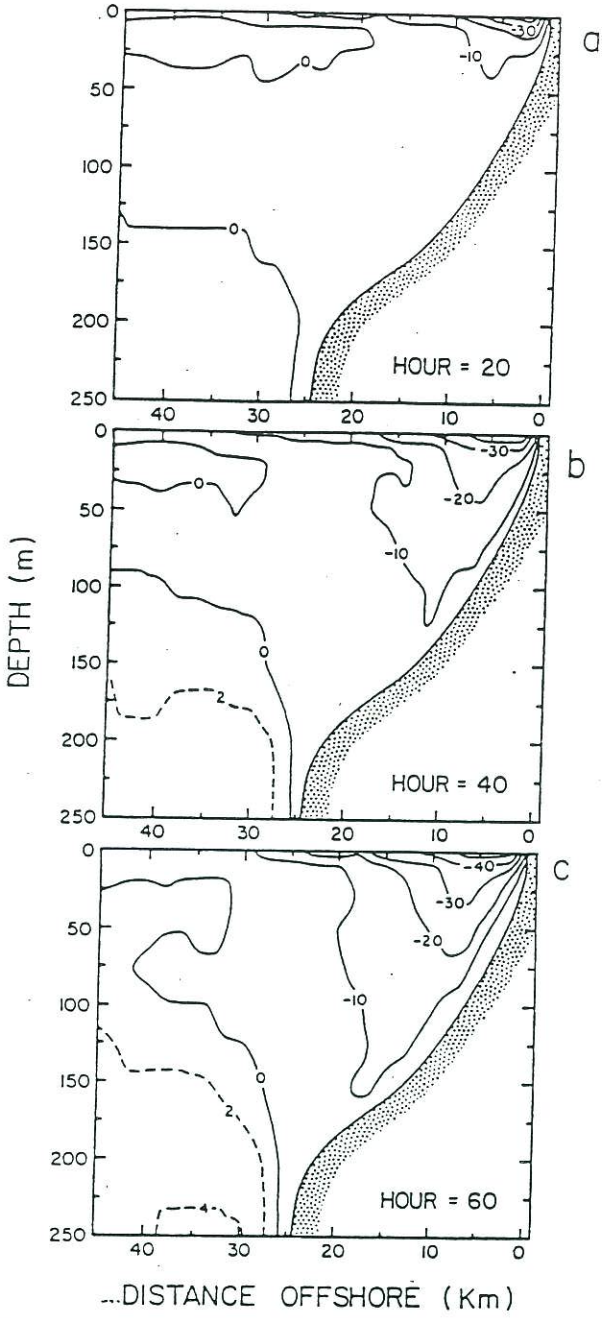


Fig. 9. Sequential patterns of alongshore velocity along a vertical section near the northern edge of the wind forcing zone (CI = 10 cm/s).

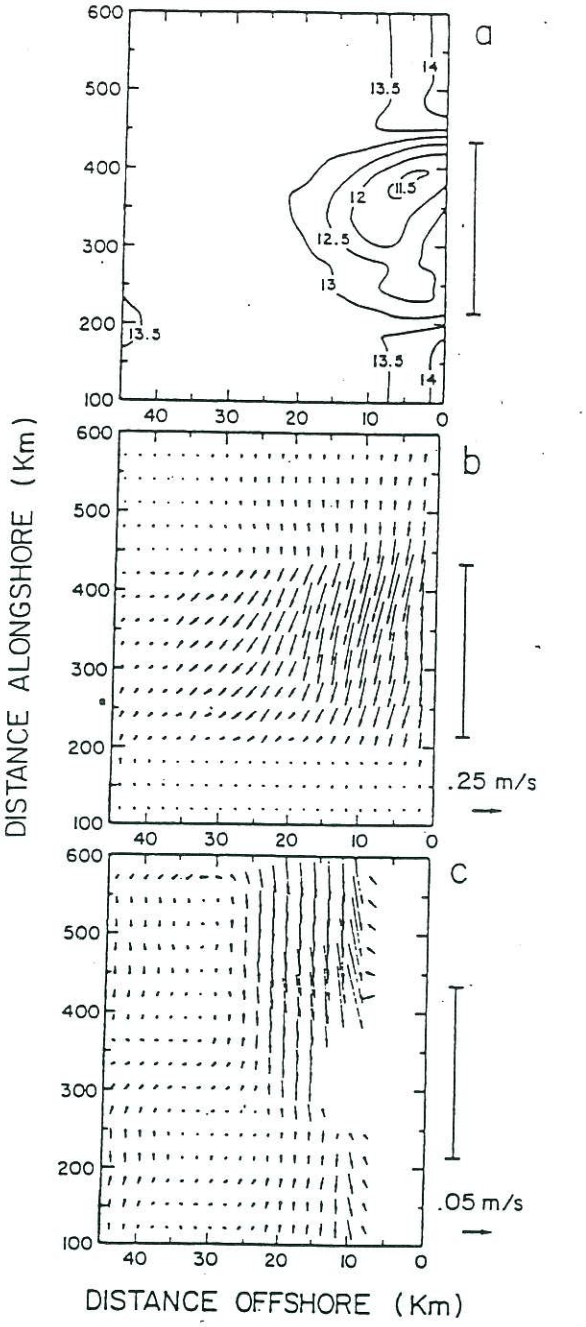


Fig. 10. The surface temperature (°C) distribution (a) and the near surface (b) and 100-m-depth (c) circulation patterns after 60 hours from the upwelling experiment with the topographic irregularity. The forcing region is marked on the right side of each figure.

Suginohara [1982], and Wang [1982] and corroborated here, the limited area wind leads to the generation of coastal trapped waves which propagate northward. The first mode waves travel at ~250 km/day. A poleward undercurrent (see Figure 9) develops below the thermocline over the slope region at ~250 m depth when the second mode coastal trapped wave arrives, travelling at ~80 km/day, from the southern edge of the forcing zone. The poleward current is apparently produced by an alongshore pressure gradient which propagates into the forcing region. It should be re-

marked that the inertial effects are responsible for offshore surface currents not always being aligned with the wind.

In the next numerical experiment the effect of alongshore variations of bottom topography is investigated. A schematic view of the topography is illustrated in Figure 2b. The length scale of the topographic irregularity is much larger than the Rossby deformation radius. Figure 10 shows the surface temperature and circulation and the 100-m-depth circulation 60 hours after the sudden onset of a limited area, upwelling favorable, 2 dyne/cm^2 wind stress. The most striking feature in Figure 10 is the enhanced upwelling which occurs over the northern edge of the topographic feature which was not present in Figure 5c. There is also some enhancement at the southern edge but to a lesser extent. The rate of upwelling is also greater when the topographic feature is included. The results obtained here are corroborated by those of Kishi and Suginohara [1975]. In both of the present numerical experiments, salinity was conserved to within machine accuracy.

While there is always some uncertainty as to the ultimate cost of a computation, the 2-1/2-day experiment with a 32 x 24, 16-level model with 144 time steps per day requires 1.0 min/day on a Cray-1S computer. The numerical experiments described here indicate that the circulation model is operational and is capable of representing the physical processes responsible for the three-dimensional behavior of the coastal ocean. By virtue of providing reliable dynamic and thermodynamic properties, it is believed that this circulation model will be useful for forecasting the behavior of coastal ocean regions.

7. Conclusions

A three-dimensional, coastal ocean circulation model has been described in detail. The model, originally designed for carrying out coastal ocean predictions, has been extended to include estuaries as well as whole basins, like the Gulf of Mexico, and to operate in both prognostic and diagnostic modes.

The vertical coordinate representation, the mode splitting technique and the open boundary conditions have been emphasized. An apparently unique feature of the circulation model is its turbulence closure submodel which should provide accurate surface and bottom mixed layer dynamics. An enhancement, now in progress, is to replace the present rectangular horizontal grid with an orthogonal, curvilinear horizontal grid to improve coastline representations.

To illustrate the utility of the circulation model, it was applied to a coastal upwelling situation. It appears from an examination of the simulation that the model yields results that share many features in common with our understanding of upwelling and coastal trapped waves. However, more qualitative comparisons between model and data/theory are necessary. It is only through these comparisons that one can understand and assess the model's predictive capabilities and limitations.

Acknowledgments. The development of the model was supported over the years by a variety of sources. In particular the authors wish to acknowledge the support provided by the Office of Sea Grant/NOAA/U.S. Department of Commerce, the New Jersey Marine Sciences Consortium, the Division of Solar Technology/U.S. Department of Energy under contract number DE-AC02-78ET20612, and the Bureau of Land Management and the Minerals Management Service of the U.S. Department of the Interior under contracts AA551-CT9-32 and AA851-CT1-67. This article was prepared with support from the Dynalysis Fund for Independent Research and Development. The authors are indebted to H. James Herring for his efforts in providing an environment conducive to research.

References

Asselin, R., Frequency filters for time integrations, Mon. Weather Rev., 100, 487-490, 1972.

Batteen, M. L., and Y.-J. Han, On the computational noise of finite-difference schemes used in ocean models, Tellus, 33, 387-396, 1981.

Beardsley, R. C., and W. C. Boicourt, On estuarine and continental-shelf circulation in the Middle Atlantic Bight, in Evolution of Physical Oceanography, edited by B. A. Warren and C. Wunsch, pg. 198-234, MIT Press, Cambridge, Mass., 1981.

Blumberg, A. F., Numerical tidal model of Chesapeake Bay, J. Hydraul. Div., 103, 1-10, 1977.

Blumberg, A. F., and L. H. Kantha, Open boundary condition for circulation models, J. Hydraul. Eng., 111, 237-255, 1985.

Blumberg, A. F., and G. L. Mellor, A whole basin model of the Gulf of Mexico. Proc. 6th Ann. Conf. on OTEC, Department of Energy, Washington, D.C., 13.15-1 to 13.15-10, 1979a.

Blumberg, A. F., and G. L. Mellor, The potential impact of three-dimensional circulation modelling on oil spill forecasting, in The Physical Behavior of Oil in the Marine Environment, Princeton University, Department of Civil Engineering, 6.1-6.18, 1979b.

Blumberg, A. F., and G. L. Mellor, A coastal ocean numerical model, in Mathematical Modelling of Estuarine Physics, Proc. Int. Symp., Hamburg, August 24-26, 1978. edited by J. Sundermann and K.-P. Holz, 203-214, Springer-Verlag, Berlin, 1980.

Blumberg, A. F., and G. L. Mellor, A numerical calculation of the circulation in the Gulf of Mexico, Dynalysis of Princeton, Report No. 66, 153 pp., 1981a.

Blumberg, A. F., and G. L. Mellor, Some results from a Gulf of Mexico circulation model, in Proc. 8th Ocean Energy Conf., Department of Energy, Washington, D.C. pp. 483-493, 1981b.

Blumberg, A. F., and G. L. Mellor, Diagnostic and prognostic numerical circulation studies of the

South Atlantic Bight, J. Geophys. Res, 88, 4579-4592, 1983.

Bryan, K., A numerical method for the study of the circulation of the world ocean, J. Comput. Phys., 4, No. 3, 347-376, 1969.

Fofonoff, N. P., Physical properties of sea-water, in The Sea, Vol. 1, edited by N. M. Hill, pp. 3-30, Interscience, New York, 1962.

Hanjalic, K., and B. E. Launder, Fully developed asymmetric flow in a plane channel, J. Fluid Mech., 52, 689, 1972.

Holland, W. R., and L. B. Lin, On the generation of mesoscale eddies and their contribution to the oceanic general circulation, I. A preliminary numerical experiment, J. Phys. Oceanogr., 5, 642-657, 1975.

Kantha, L. H., G. L. Mellor, and A. F. Blumberg, A diagnostic calculation of the general circulation in the South Atlantic Bight, J. Phys. Oceanogr., 12, 805-819, 1982.

Kerr, C. L., and A. F. Blumberg, An analysis of a local second-moment conserving quasi-Lagrangian scheme for solving the advection equation, J. Comput. Phys., 32, 1-9, 1979.

Kishi, M. J., and N. Suginohara, Effects of longshore variation of coastline geometry and bottom topography on coastal upwelling in a two-layer model, J. Oceanogr.Soc. Japan, 31, 48-50, 1975.

Leendertse, J. J., R. C. Alexander, and S.-K. Liu, A three-dimensional model for estuaries and coastal seas, Volume I, Principles of Computation, The Rand Corp., Santa Monica, Calif., R-1417-OWRR, 1973.

Lilly, D. K., On the computational stability of numerical solutions of time-dependent nonlinear geophysical fluid dynamics problems, Mon. Weather Rev., 93, 11-26, 1965.

Madala, R. V., and S. A. Piacsek, A semi-implicit numerical model for baroclinic oceans, J. Comput. Phys., 23, 167-178, 1977.

Mellor, G. L., Analytic prediction of the properties of stratified planetary surface layers, J. Atmos. Sci., 30, 1061-1069, 1973.

Mellor, G. L., and A. F. Blumberg, Modelling vertical and horizontal diffusivities and the sigma coordinate system, Mon. Weather Rev., 113, 1379-1383, 1985.

Mellor, G. L., and H. J. Herring, A survey of the mean turbulent field closure models, AIAA J., 11, 590-599, 1973.

Mellor, G. L., and T. Yamada, A hierarchy of turbulence closure models for planetary boundary layers, J. Atmos. Sci., 31, 1791-1896, 1974.

Mellor, G. L., and T. Yamada, Development of a turbulence closure model for geophysical fluid problems, Rev. Geophys. Space Phys., 20, No. 4, 851-875, 1982.

Philander, S.G.H., and J.-H. Yoon, Eastern boundary currents and coastal upwelling, J. Phys. Oceanogr., 12, No. 8, 862-879, 1982.

Phillips, N. A., A coordinate system having some special advantages for numerical forecasting, J. Meteorol., 14, 184-185, 1957.

Richtmyer, R. D., and K. W. Morton, Difference Methods for Initial-Value Problems, Second Edition, Interscience, New York, 1967.

Simons, T. J., Verification of numerical models of Lake Ontario, Part I. Circulation in spring and early summer, J. Phys. Oceanogr., 4, 507-523, 1974.

Smagorinsky, J., General circulation experiments with the primitive equations, I. The basic experiment, Mon. Weather Rev., 91, 99-164, 1963.

Suginohara, N., Coastal upwelling: Onshore-offshore circulation, equatorward coastal jet and poleward undercurrent over a continental shelf-slope, J. Phys. Oceanogr., 12, No. 3, 272-284, 1982.

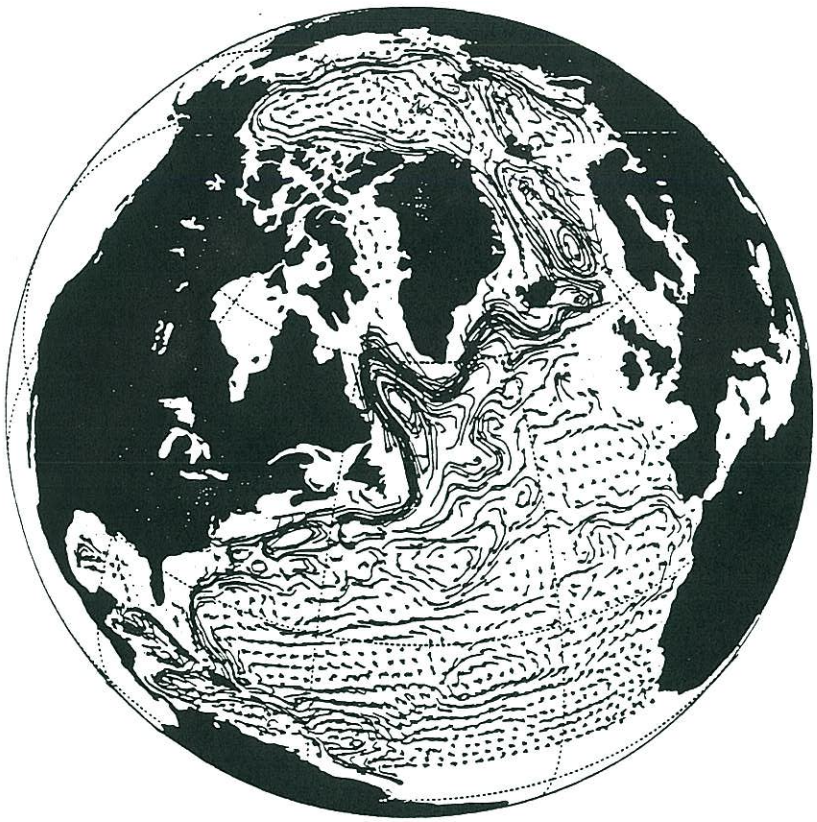
Wang, D.-P., Development of a three-dimensional, limited-area (island) shelf circulation model, J. Phys. Oceanogr., 12, 605-617, 1982.

Weatherly, G., and P. J. Martin, On the structure and dynamics of the ocean bottom boundary layer, J. Phys. Oceanogr., 8, 557-570, 1978.

USERS GUIDE for

A THREE-DIMENSIONAL, PRIMITIVE EQUATION, NUMERICAL OCEAN MODEL

George L. Mellor
Atmospheric and Oceanic Sciences Program
Princeton University
Princeton, NJ 08540



This revision: JUNE 1996

About This Revision This version of the users guide recognizes changes that have ocurred since 1991. The code itself incorporates some recent changes. The Fortran names, TMEAN, SMEAN have been changed (globally) to TCLIM, SCLIM in oder to distiquish the function and treatment of these variables from that of RMEAN. The names , TRNU, TNRV, have been changed to DRX2D, DRY2D and the names, ADVUU, ADVVV, to ADX2D, ADY2D to more clearly indicate their functions. Instead of a wind driven closed basin, pom.f now solves the problem of the flow through a channel which includes an island or a seamount at the center of the domain. Thus, subroutine BCOND contains active open boundary conditions. These illustrative boundary conditions, however, are one set of many possibilities and, consequently, open boundary conditions for regional models pose difficult choices for users of the model. This revision contains a fuller discussion of open boundary conditions in section 16.

As of this revision date, there are about 200 POM users of record.

Sponsor Acknowledgment: The development and application of the program has had many sponsors since 1977. They include the Geophysical Fluid Dynamics Laboratory/NOAA, Princeton University, Sea Grant/NOAA through the New Jersey Marine Sciences Consortium, the Department of Energy, Minerals Management Services/DOI, the National Ocean Services/NOAA, the Institute of Naval Oceanography and the Office of Naval Research/DOD.

Title Page Illustration: North Atlantic velocity field on the 32.45 potential density surface. Courtesy Dr. Sirpa Häkkinen.

CONTENTS

1. INTRODUCTION

This report is documentation for a numerical ocean model created by Alan Blumberg and me around 1977. Subsequent contributions were made by Leo Oey, Jim Herring, Lakshmi Kantha and Boris Galperin and others. In recent years Tal Ezer has been an important force in research using the model and in helping others use it. Institutionally, the model was developed and applied to oceanographic problems in the Atmospheric and Oceanic Sciences Program of Princeton University, the Geophysical Fluid Dynamics Laboratory of NOAA and Dynalysis of Princeton. Many sponsors, as acknowleged above, have supported the effort. Papers that either describe the numerical model (Blumberg and Mellor, 1987) or made use of the model are contained in the Reference Section and a more complete list is available on the POM home page at http://www.aos.princeton.edu/wwwpublic/htdocs.pom/.

The model is oftentimes referenced as the Princeton Ocean Model (POM). The principal attributes of the model are as follows:

o It contains an imbedded second moment turbulence closure sub-model to provide vertical mixing coefficients.
o It is a sigma coordinate model in that the vertical coordinate is scaled on the water column depth.
o The horizontal grid uses curvilinear orthogonal coordinates and an "Arakawa C" differencing scheme.
o The horizontal time differencing is explicit whereas the vertical differencing is implicit. The latter eliminates time constraints for the vertical coordinate and permits the use of fine vertical resolution in the surface and bottom boundary layers.
o The model has a free surface and a split time step. The external mode portion of the model is two-dimensional and uses a short time step based on the CFL condition and the external wave speed. The internal mode is three-dimensional and uses a long time step based on the CFL condition and the internal wave speed.
o Complete thermodynamics have been implemented.

The turbulence closure sub-model is one that I introduced (Mellor, 1973) and then was significantly advanced in collaboration with Tetsuji Yamada (Mellor and Yamada,1974; Mellor and Yamada,1982). It is often cited in the literature as the Mellor-Yamada turbulence closure model (but, it should be noted that the model is based on

turbulence hypotheses by Rotta and Kolmogorov which we extended to stratified flow cases). Here, the Level 2.5 model is used together with a prognostic equation for the turbulence macroscale. The closure model is contained in subroutines PROFQ and ADVQ. A list of papers pertaining to the closure model is also included in the Reference section.

By and large, the turbulence model seems to do a fair job simulating mixed layer dynamics although there have been indications that calculated mixed layer depths are a bit too shallow (Martin, 1985). At least a part of the deficit is due to internal wave velocity shears (Mellor, 1989). Also, wind forcing may be spatially smoothed and temporally smoothed. It is known that the latter process will reduce mixed layer thicknesses (Klein, 1980). Further study is required to quantify these effects.

The sigma coordinate system is probably a necessary attribute in dealing with significant topographical variability such as that encountered in estuaries or over continental shelf breaks and slopes. Together with the turbulence sub-model, the model produces realistic bottom boundary layers which are important in coastal waters (Mellor, 1985) and in tidally driven estuaries (Oey et al., 1985a, b) which the model can simulate since it does have a free surface. More recently, we find that bottom boundary layers are important for deep water formation processes (Zavatarelli and Mellor, 1995; Jungclaus and Mellor, 1996; Baringer and Price, 1996) and, possibly for the maintenance of the baroclinicity of oceans basins (Mellor and Wang, 1996).

The horizontal finite difference scheme is staggered and, in the literature, has been called an Arakawa C-grid. The horizontal grid is a curvilinear coordinate system, or as a special case, a rectilinear coordinate system may be easily implemented. The advection, horizontal diffusion and, in the case of velocity, the pressure gradient and Coriolis terms are contained in subroutines ADVT, ADVQ, ADVCT, ADVU, ADVV and ADVAVE. The horizontal differencing could be changed without affecting the overall logic of the program or the remaining subroutines. The vertical diffusion is handled in subroutines, PROFT, PROFQ, PROFU and PROFV.

The specific program that is now supplied to outside users (as of June 1996) simulates the flow, east to west across a seamount with a prescribed vertical temperature stratification, constant salinity, zero surface heat and salinity flux and a zero wind stress distribution although wind stress may be easily applied. The program should run with no additional data requirements. The open boundary conditions specified in SUBROUTINE BCOND for this problem are a sampling of many possible open boundary conditions. I leave it to users to invent their own problems, defined by topography, horizontal grid (rectilinear, where DX(I, J) is specified as a function of I and DY(I, J) as a function of J, or a more general orthogonal curvilinear grid), vertical sigma grid and boundary

conditions. Users may need to alter PROGRAM MAIN and SUBROUTINE BCOND; in principal, there should be no need to alter any of the other subroutines.

The present program code is written in standard FORTRAN 77. There are other versions in existence, but we only support and maintain the single version.

Provision has been made so that the 2-D (external mode) portion of the model can be run *cum sole*. In this case, the bottom shear stress, normally a consequence of the 3-D calculation and the turbulence mixing coefficient, is replaced by a quadratic drag relation. The code may also be run in a diagnostic mode where the thermodynamic properties are invariant in time.

Users will need to write their own code to set up initial conditions and lateral and surface boundary conditions. We can, however, supply simple subroutines that convert data for a constant z-level coordinate system to a sigma coordinate system and vice versa.

To acces pom.f and other files through Internet, type **ftp ftp.gfdl.gov**; when prompted for your name, type **anonymous**; when prompted for a password, type your iternet address; after receiving a guest login ok, type **cd pub/glm**. You may list filenames with the **ls** command. You may download with the command **get filename**. Type **quit** to terminate.

2. THE BASIC EQUATIONS

The basic equations have been cast in a bottom following, sigma coordinate system which is illustrated in Figure 1. The reader is referred to Phillips (1957) or Blumberg and Mellor (1980,1987) for a derivation of the sigma coordinate equations which are based on the transformation,

$$x^* = x,\ y^* = y,\ \sigma = \frac{z-\eta}{H+\eta},\ t^* = t \qquad \text{(1a, b, c, d)}$$

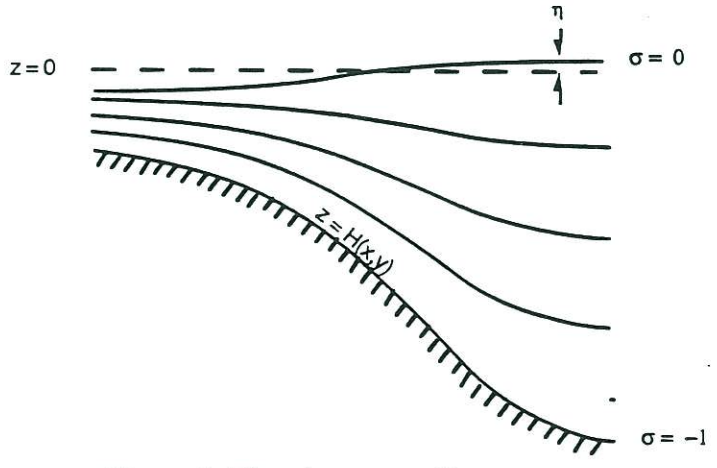


Figure 1. The sigma coordinate system.

where x,y,z are the conventional cartesian coordinates; $D \equiv H + \eta$ where $H(x, y)$ is the bottom topography and $\eta(x, y, t)$ is the surface elevation. Thus, σ ranges from $\sigma = 0$ at $z = \eta$ to $\sigma = -1$ at $z = -H$. After conversion to sigma coordinates and deletion of the asterisks, the basic equations may be written (in horizontal cartesian coordinates),

$$\frac{\partial DU}{\partial x} + \frac{\partial DV}{\partial y} + \frac{\partial \omega}{\partial \sigma} + \frac{\partial \eta}{\partial t} = 0 \tag{2}$$

$$\frac{\partial UD}{\partial t} + \frac{\partial U^2 D}{\partial x} + \frac{\partial UVD}{\partial y} + \frac{\partial U\omega}{\partial \sigma} - fVD + gD\frac{\partial \eta}{\partial x} + \frac{gD^2}{\rho_o}\int_\sigma^o \left[\frac{\partial \rho'}{\partial x} - \frac{\sigma'}{D}\frac{\partial D}{\partial x}\frac{\partial \rho'}{\partial \sigma'}\right] d\sigma' = \frac{\partial}{\partial \sigma}\left[\frac{K_M}{D}\frac{\partial U}{\partial \sigma}\right] + F_x \tag{3}$$

$$\frac{\partial VD}{\partial t} + \frac{\partial UVD}{\partial x} + \frac{\partial V^2 D}{\partial y} + \frac{\partial V\omega}{\partial \sigma} + fUD + gD\frac{\partial \eta}{\partial y} + \frac{gD^2}{\rho_o}\int_\sigma^o \left[\frac{\partial \rho'}{\partial y} - \frac{\sigma'}{D}\frac{\partial D}{\partial y}\frac{\partial \rho'}{\partial \sigma'}\right] d\sigma' = \frac{\partial}{\partial \sigma}\left[\frac{K_M}{D}\frac{\partial V}{\partial \sigma}\right] + F_y \tag{4}$$

$$\frac{\partial TD}{\partial t} + \frac{\partial TUD}{\partial x} + \frac{\partial TVD}{\partial y} + \frac{\partial T\omega}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left[\frac{K_H}{D}\frac{\partial T}{\partial \sigma}\right] + F_T - \frac{\partial R}{\partial z} \tag{5}$$

$$\frac{\partial SD}{\partial t} + \frac{\partial SUD}{\partial x} + \frac{\partial SVD}{\partial y} + \frac{\partial S\omega}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left[\frac{K_H}{D}\frac{\partial S}{\partial \sigma}\right] + F_S \tag{6}$$

$$\frac{\partial q^2 D}{\partial t} + \frac{\partial Uq^2 D}{\partial x} + \frac{\partial Vq^2 D}{\partial y} + \frac{\partial \omega q^2}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left[\frac{K_q}{D}\frac{\partial q^2}{\partial \sigma}\right] + \frac{2K_M}{D}\left[\left(\frac{\partial U}{\partial \sigma}\right)^2 + \left(\frac{\partial V}{\partial \sigma}\right)^2\right] + \frac{2g}{\rho_o}K_H\frac{\partial \tilde{\rho}}{\partial \sigma} - \frac{2Dq^3}{B_1 \ell} + F_q \tag{7}$$

$$\frac{\partial q^2 \ell D}{\partial t} + \frac{\partial Uq^2 \ell D}{\partial x} + \frac{\partial Vq^2 \ell D}{\partial y} + \frac{\partial \omega q^2 \ell}{\partial \sigma} = \frac{\partial}{\partial s}\left[\frac{K_q}{D}\frac{\partial q^2 \ell}{\partial \sigma}\right] + E_1 \ell\left(\frac{K_M}{D}\left[\left(\frac{\partial U}{\partial \sigma}\right)^2 + \left(\frac{\partial V}{\partial \sigma}\right)^2\right] + E_3\frac{g}{\rho_o}K_H\frac{\partial \tilde{\rho}}{\partial \sigma}\right)\tilde{W} + F_\ell \tag{8}$$

Definitions of the variables are contained in section 3. Note that ω is the transformed vertical velocity ; physically, ω is the velocity component normal to sigma surfaces. The so-called wall proximity function is prescribed according to $\tilde{W} = 1 + E_2(\ell / kL)$ where $L^{-1} = (\eta - z)^{-1} + (H - z)^{-1}$. Also, $\partial\tilde{\rho} / \partial\sigma \equiv \partial\rho / \partial\sigma - c_s^{-2}\partial p / \partial\sigma$ (see discussion of static stability in Appendix A) where c_s is the speed of sound. Note that T is potential temperature (see Appendix A).

In equations (3) and (4), RMEAN shound be subtracted from ρ to form ρ' before the integration is carried out in subroutine BAROPG. RMEAN is generally the initial density field which is area averaged and then transferred to sigma coordinates in the exact same way as the initial density field.

The horizontal viscosity and diffusion terms are defined according to:

$$F_x \equiv \frac{\partial}{\partial x}(H\tau_{xx}) + \frac{\partial}{\partial y}(H\tau_{xy}) \tag{9a}$$

$$F_y \equiv \frac{\partial}{\partial x}(H\tau_{xy}) + \frac{\partial}{\partial y}(H\tau_{yy}) \tag{9b}$$

where

$$\tau_{xx} = 2A_M \frac{\partial U}{\partial x}, \quad \tau_{xy} = \tau_{yx} = A_M\left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}\right), \quad \tau_{yy} = 2A_M \frac{\partial V}{\partial y} \tag{10a,b,c}$$

Also,

$$F_\phi \equiv \frac{\partial}{\partial x}(Hq_x) + \frac{\partial}{\partial y}(Hq_y) \tag{11}$$

where

$$q_x \equiv A_H \frac{\partial \phi}{\partial x}, \quad q_y \equiv A_H \frac{\partial \phi}{\partial y} \tag{12a,b}$$

and where ϕ represents T, S, q^2 or $q^2\ell$. It should be noted that these horizontal diffusion terms are not what one would obtain by transforming the conventional forms to the sigma coordinate system. Justification for the present forms will be found in Mellor and Blumberg (1985) and relate to the fact that we wish to maintain a valid bottom boundary layer simulation in the face of horizontal diffusion which may be large. The penalty for this is that (12a,b) in sigma coordinates can introduce vertical fluxes even when isotherms and isohalines are flat in cartesian coordinates. The remedy for this is, first, the use of a

Smagorinsky diffusivity so that, at least when velocities are small or nil, so are the values of q_x and q_y. The second remedy is that, before executing (12a, b) for temperature or salinity, we first subtract T_{CLIM} and S_{CLIM} which are "climatologies" of T and S. The latter may be true climatologies (e.g.; Levitus) or approximations such as temperature and salinities which are area averaged prior to transfer to sigma coordiates (in which case, they are treated the same as ρ_{MEAN}). If something like a Levitus climatology is used, then most of the vertical component of the diffusion is removed; furthermore, the diffusion terms tend to drive the scalars back to climatology rather than to a horizontally homogeneous state as in the case of z - level models. As resolution improves, the whole issue disappears. In the meantime, one has a three-dimensional model that can model bottom boundary layers. The bottom boundary layer is important in tidally driven regions, in wind driven coastal regions and according to Mellor and Wang (1996), in deep ocean basins.

In (9a, b) and (11), H is used in place of D for the small algorithmic simplication it offers for terms whose physical significance is questionable.

Vertical Boundary Conditions.

The vertical boundary conditions for (2) are

$$\omega(0) = \omega(-1) = 0 \tag{13a,b}$$

The boundary conditions for (3) and (4) are

$$\frac{K_M}{D}\left(\frac{\partial U}{\partial \sigma}, \frac{\partial V}{\partial \sigma}\right) = -\left(<\omega u(0)>, <\omega v(0)>\right), \sigma \rightarrow 0 \tag{14a,b}$$

where the right hand side of (14a,b) is the input values of the surface wind stress (divided by ρ_o), and

$$\frac{K_M}{D}\left(\frac{\partial U}{\partial \sigma}, \frac{\partial V}{\partial \sigma}\right) = C_z\left[U^2 + V^2\right]^{1/2}(U,V), \sigma \rightarrow -1 \tag{14c,d}$$

where

$$C_z = MAX\left[\frac{k^2}{\left[\ln\left\{(1+\sigma_{kb-1})H / z_o\right\}\right]^2}, 0.0025\right] \tag{14e}$$

$k = 0.4$ is the von Karman constant and z_o is the roughness parameter. Equations (14c,d,e) can be derived by matching the numerical solution to the "law of the wall". Numerically, they are applied to the first grid points nearest the bottom. Where the bottom is not well resolved, $(1+\sigma_{kb-1})H/z_o$ is large and (14e) reverts to an ordinary drag coefficient formulation. The boundary conditions on (5) and (6) are

$$\frac{K_H}{D}\left(\frac{\partial T}{\partial \sigma}, \frac{\partial S}{\partial \sigma}\right) = -\left(<\omega\theta(0)>\right) \ , \ \sigma \rightarrow 0 \qquad (15a,b)$$

$$\frac{K_H}{D}\left(\frac{\partial T}{\partial \sigma}, \frac{\partial S}{\partial \sigma}\right) = 0 \ , \qquad \sigma \rightarrow -1 \qquad (15c,d)$$

The boundary conditions for (7) and (8) are

$$\left(q^2(0), q^2\ell(0)\right) = \left(B_1^{2/3} u_\tau^2(0),\ 0\right) \qquad (16a,b)$$

$$\left(q^2(-1),\ q^2\ell(-1)\right) = \left(B_1^{2/3}\, u_\tau^2(-1),\ 0\right) \qquad (16c,d)$$

where B_1 is one of the turbulence closure constants and u_τ is the friction velocity at the top or bottom as denoted.

The Vertically Integrated Equations

The equations, governing the dynamics of coastal circulation, contain fast moving external gravity waves and slow moving internal gravity waves. It is desirable in terms of computer economy to separate the vertically integrated equations (external mode) from the vertical structure equations (internal mode). This technique, known as mode splitting (Simons, 1974; Madala and Piacsek, 1977) permits the calculation of the free surface elevation with little sacrifice in computational time by solving the velocity transport separately from the three-dimensional calculation of the velocity and the thermodynamic properties.

The velocity transport, external mode equations are obtained by integrating the internal mode equations over the depth, thereby eliminating all vertical structure. Thus, by integrating Equation (2) from $\sigma = -1$ to $\sigma = 0$ and using the boundary conditions (13a,b), an equation for the surface elevation can be written as

$$\frac{\partial \eta}{\partial t} + \frac{\partial \overline{U} D}{\partial x} + \frac{\partial \overline{V} D}{\partial y} = 0 \qquad (17)$$

After integration, the momentum equations, (3) and (4), become

$$\frac{\partial \overline{U}D}{\partial t}+\frac{\partial \overline{U}^2D}{\partial x}+\frac{\partial \overline{U}\overline{V}D}{\partial y}-\tilde{F}_x-f\overline{V}D+gD\frac{\partial \eta}{\partial x}=-<\omega u(0)>+<\omega u(-1)> + G_x-\frac{gD}{\rho_o}\int_{-1}^{o}\int_{\sigma}^{o}\left[D\frac{\partial \rho'}{\partial x}-\frac{\partial D}{\partial x}\sigma'\frac{\partial \rho'}{\partial \sigma}\right]d\sigma'\,d\sigma \tag{18}$$

$$\frac{\partial \overline{V}D}{\partial t}+\frac{\partial \overline{U}\overline{V}D}{\partial x}+\frac{\partial \overline{V}^2D}{\partial y}-\tilde{F}_y+f\overline{U}D+gD\frac{\partial \eta}{\partial y}=-<\omega v(0)>+<\omega v(-1)> + G_y-\frac{gD}{\rho_o}\int_{-1}^{o}\int_{\sigma}^{o}\left[D\frac{\partial \rho'}{\partial y}-\frac{\partial D}{\partial y}\sigma'\frac{\partial \rho'}{\partial \sigma}\right]d\sigma'\,d\sigma \tag{19}$$

The overbars denote vertically integrated velocities such as

$$\overline{U}\equiv\int_{-1}^{o}U\,d\sigma\,. \tag{20}$$

The wind stress components are $-<\omega u(0)>$ and $-<\omega u(0)>$, and the bottom stress components are $-<\omega u(-1)>$ and $-<\omega u(-1)>$. The quantities $\tilde{F}_x$ and $\tilde{F}_y$ are defined according to

$$\tilde{F}_x=\frac{\partial}{\partial x}\left[H2\overline{A}_M\frac{\partial \overline{U}}{\partial x}\right]+\frac{\partial}{\partial y}\left[H\overline{A}_M\left(\frac{\partial \overline{U}}{\partial y}+\frac{\partial \overline{V}}{\partial x}\right)\right] \tag{21a}$$

and

$$\tilde{F}_y=\frac{\partial}{\partial y}\left[H2\overline{A}_M\frac{\partial \overline{V}}{\partial y}\right]+\frac{\partial}{\partial x}\left[H\overline{A}_M\left(\frac{\partial \overline{U}}{\partial y}+\frac{\partial \overline{V}}{\partial x}\right)\right] \tag{21b}$$

The so-called dispersion terms are defined according to

$$G_x=\frac{\partial \overline{U}^2D}{\partial x}+\frac{\partial \overline{U}\overline{V}D}{\partial y}-\tilde{F}_x-\frac{\partial \overline{U^2}D}{\partial x}-\frac{\partial \overline{UV}D}{\partial y}+\overline{F}_x \tag{22a}$$

$$G_y=\frac{\partial \overline{U}\overline{V}D}{\partial x}+\frac{\partial \overline{V}^2D}{\partial y}-\tilde{F}_y-\frac{\partial \overline{UV}D}{\partial x}-\frac{\partial \overline{V^2}D}{\partial y}+\overline{F}_y \tag{22b}$$

Note that, if A_M is constant in the vertical, then the last two terms in (22a) and (22b) cancel. However, we account for possible vertical variability in the horizontal diffusivity; such is the case when a Smagorinsky type diffusivity is used. As detailed below, all of the terms on the right side of (18) and (19) are evaluated at each internal time step and then

held constant throughout the many external time steps. If the external mode is executed *cum sole*, then $G_x = G_y = 0$.

The Smagorinsky Diffusivity

We generally use the Smagorinsky diffusivity for horizontal diffusion although a constant or biharmonic diffusion can and has been used instead. The Smagorinsky formula is,

$$A_M = C\Delta x\Delta y \frac{1}{2}\left|\nabla \mathbf{V} + (\nabla \mathbf{V})^T\right|$$

where $\left|\nabla \mathbf{V} + (\nabla \mathbf{V})^T\right| /2 = [(\partial u / \partial x)^2 + (\partial v / \partial x + \partial u / \partial y)^2 / 2 + (\partial v / \partial y)^2]^{1/2}$. Values of C in the range, 0.10 to 0.20 seem to work well, but, if the grid spacing is small enough (Oey *et al*, 1985a,b), C can be nil. An advantage of the Smagorinsky is that C is non-dimensional; related advantages are that A_M decreases as resolution improves and that A_M is small if velocity gradients are small.

3. FORTRAN SYMBOLS

In the following table, we list the FORTRAN symbols followed by their corresponding analytical symbols in parentheses and a brief description of the symbols. Not explicitly tabulated are the suffixes B, blank and F which are appended to many of the variables to denote the time levels n - 1, n and n + 1 respectively.

Indices

I, J (i, j)	horizontal grid indexes
IM, JM	outer limits of I and J
K (k)	vertical grid index; K = 1 at the top and K = KB at the bottom
IINT (n)	internal mode time step index
IEXT	external mode time step index

Constants

DTE (Δt_E)	external mode time step, (s)
DTI (Δt_I)	internal mode time step, (s)
EXTINC	short wave extinction coefficient, (m^{-1})
IEND	total internal mode time steps

IPRINT	the interval in IINT at which variables are printed
ISPLIT	DTI/DTE
MODE	if MODE = 2, a 2-D calculation is performed if MODE = 3, a 3-D prognostic calculation is performed if MODE = 4, a 3-D diagnostic calculation is performed
SMOTH (α)	parameter in the temporal smoother
TPRNI (A_H/A_M)	inverse, horizontal, turbulence Prandtl number
TR	short wave surface transmission coefficient

One-dimensional Arrays

Z(σ)	sigma coordinate which spans the domain, Z = 0 (surface) to Z = -1 (bottom)
ZZ	sigma coordinate, intermediate between Z
DZ($\delta\sigma$)	= Z(K)–Z(K+1)
DZZ	= ZZ(K)–ZZ(K+1)

Two-dimensional Arrays

ART, ARU, ARV	cell areas centered on the variables, T, U and V respectively (m^2)
AAM (A_M)	horizontal kinematic viscosity ($m^2\ s^{-1}$)
AAH (A_H)	horizontal heat diffusivity ($m^2 s^{-1}$)
ADVUA, ADVVA	sum of the second, third and fourth terms in equations (18,19)
ADX2D, ADY2D	vertical integrals of ADVX, ADVY; also the sum of the fourth, fifth and sixth terms in equations (22a,b)
COR (f)	the Coriolis parameter (s^{-1})
CURV2D	the vertical average of CURV
DUM	Mask for the u component of velocity; = 0 over land; = 1 over water
DVM	Mask for the v component of velocity; = 0 over land; = 1 over water
FSM	Mask for scalar variables; = 0 over land; = 1 over water
DX (h_x or δx)	grid spacing (m)
DY (h_y or δy)	grid spacing (m)

EL (η)	the surface elevation as used in the external mode (m)
ET (η)	the surface elevation as used in the internal mode and derived from EL (m)
EG (η)	the surface elevation also used in theinternal mode for the pressure gradient and derived from EL (m)
D (D)	= H + EL (m)
DT (D)	= H + ET (m)
DRX2D, DRX2D	vertical integrals of DRHOX and DRHOY
H (H)	the bottom depth (m)
SWRAD	short wave radiation incident on the ocean surface (m s^{-1}K)
UA, VA ($\bar{U},\bar{V}$)	vertical mean of U,V (m s^{-1})
UT, VT ($\bar{U},\bar{V}$)	UA,VA time averaged over the interval, DT = DTI (m s^{-1})
WUSURF, WVSURF	($<\omega u(0)>$; $<\omega v(0)>$) momentum fluxes at the surface (m^2s^{-2})
WUBOT, WUBOT	($<\omega u(-1)>$, $<\omega v(-1)>$) momentum fluxes at the bottom (m^2s^{-2})
WTSURF, WSSURF	($<\omega\theta(0)>$, $<\omega s(0)>$) temperature and salinity fluxes at the surface (ms^{-1} K, ms^{-1} psu)

Three-dimensional Arrays

ADVX, ADVY	horizontal advection and diffusion terms in equations (3) and (4)
CURV ($\tilde{f}$)	curvature terms; see equation (28)
L (ℓ)	turbulence length scale
KM (K_M)	vertical kinematic viscosity (m^2s^{-1})
KH (K_H)	vertical diffusivity (m^2s^{-1})
DRHOX $\left(gDh_y\rho_o^{-1}\left[-D\int_\sigma^0 \delta_x\rho'\,\delta\sigma' + \delta_x D\int_\sigma^0 \sigma'\delta\rho'\right]\right)$	x-component of the internal baroclinic pressure gradient subtract RMEAN from density before integrating
DRHOY $\left(gDh_x\rho_o^{-1}\left[-D\int_\sigma^0 \delta_y\rho'\,\delta\sigma' + \delta_y D\int_\sigma^0 \sigma'\delta\rho'\right]\right)$	y-component of the internal baroclinic pressure gradient subtract RMEAN from density before integrating
RAD (R)	short wave radiation flux (ms^{-1}K). Sign is the same as WTSURF

Q2 (q^2)	twice the turbulence kinetic energy (m^2s^{-2})
Q2L ($q^2 \ell$)	Q2 x the turbulence length scale (m^3s^{-2})
T (Θ)	potential temperature (K)
S (S)	salinity (psu)
RHO (ρ/ρ_0 - 1.025)	density (non-dim.)
U, V (U, V)	horizontal velocities ($m\ s^{-1}$)
W (ω)	sigma coordinate vertical velocity ($m\ s^{-1}$)
RMEAN	density field which is horizontally averaged before transfer to sigma coordinates.
TCLIM	a stationary temperature field which approximately has the same vertical structure as T.
SCLIM	a stationary salinity field which approximately has the same vertical structure as S.

The variables *UF* and *VF* are used to denote the $n+1$ time level for U and V respectively. However, in order to save memory they are also used to represent the $n+1$ time level for T and S and for Q2 and Q2L respectively. As soon as UF, VF are calculated for each representation, the time level is reset.

4. THE NUMERICAL SCHEME

Figure 2 is the flow chart for the program in simplified form. The external mode calculation is contained in PROGRAM MAIN.

External-Internal Mode Interaction.

The external mode calculation in MAIN results in updates for surface elevation, EL, and the vertically averaged velocities, UA, VA. The internal mode calculation results in updates for U,V,T,S and the turbulence quantities.

Fig. 3 illustrates the time stepping process for the external and internal mode. Assume everything is known at t^{n-1} and t^n (the previous leap frog time step having just been completed). Integrals involving the baroclinic forcing and the advective terms are supplied to the external mode along with the bottom stress, a process which is labeled "Feedback" in Fig. 3; their values are held constant during $t^n < t < t^{n+1}$. The external mode "leap frogs" many times, with the time step, DTE, until $t = t^{n+1}$. The vertical and

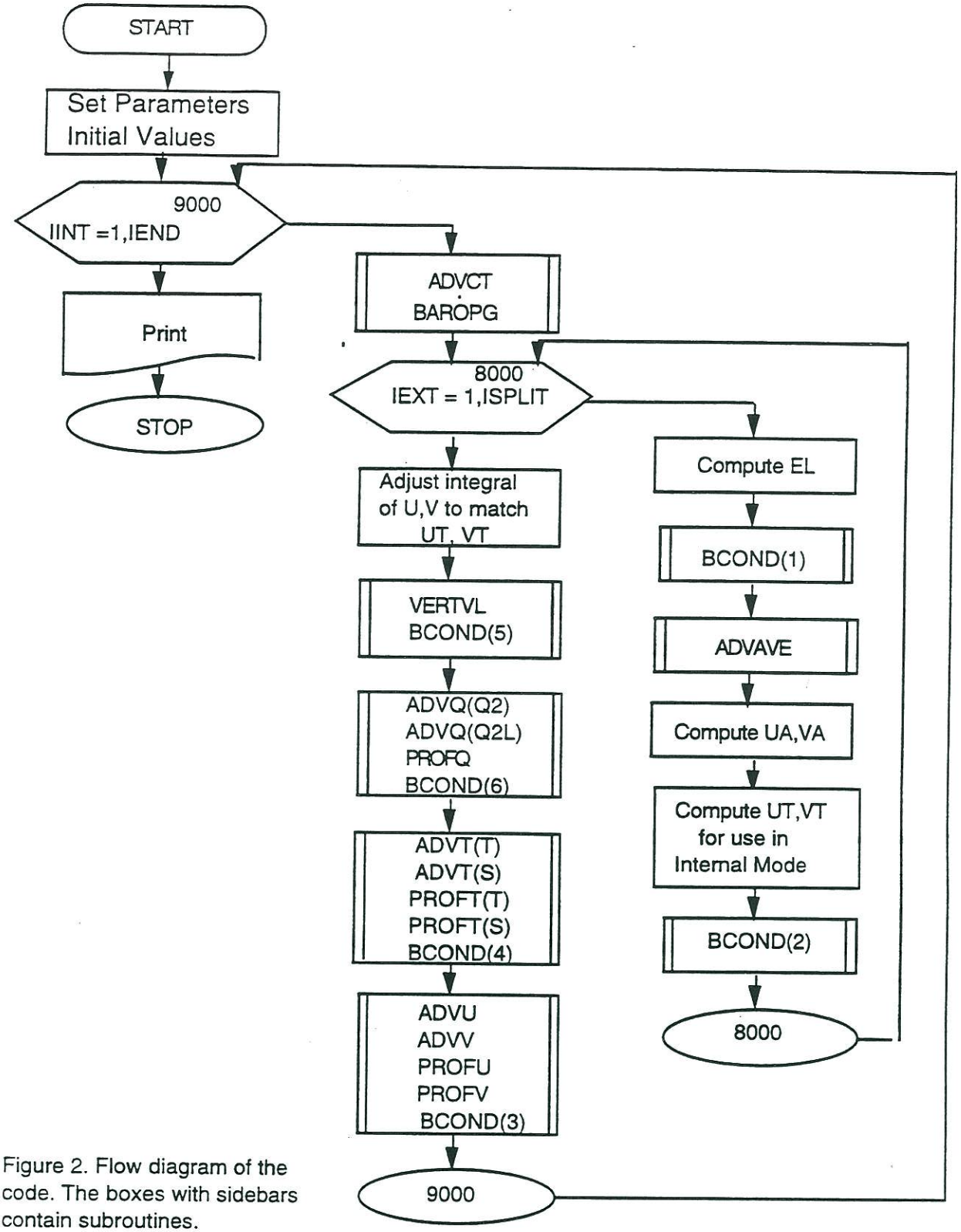


Figure 2. Flow diagram of the code. The boxes with sidebars contain subroutines.

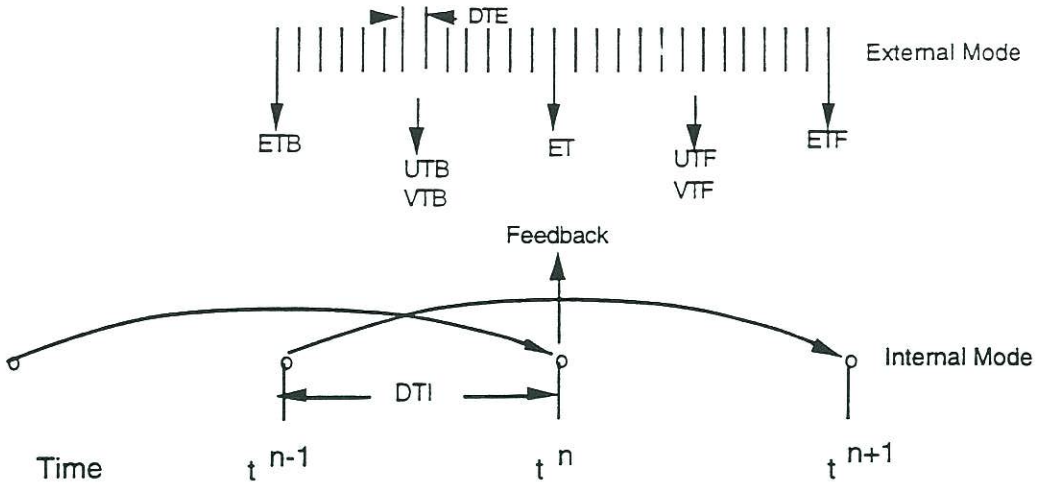


Figure 3. A simplified illustration of the interaction of the External Mode and the Internal Mode. The former uses a short time step, DTE, whereas the latter uses a long time step, DTI. The external mode primarily provides the surface elevation to the internal mode whereas, as symbolized by "Feedback", the internal mode provides intergrals of momentum advection, density integrals and bottom stress to the external mode.

time averaged velocities, UTF, VTF, and those from the previous time step, UTB,VTB, are time averages of the external variables, UA,VA. The internal and external modes have different truncation errors so that the vertical integrals of the internal mode velocity may depart slightly from (UA,VA) during the course of a long integration. We therefore adjust the internal velocities, U,V, so that their vertical integrals are the mean of UTF,VTF and UTB,VTB. Care is taken to relate ETF to ELF so that together with ETB, saved from a previous time step, the internal velocities and ETF and ETB correctly satisfy the continuity equation, (17). Otherwise, the sigma coordinate equations for T, S will not be conservative.

Aside from the above, numerically important details, 0.5*(EGF + EGB) is used to obtain the elevation gradients for the internal mode "leap frog" from t^{n+1} to t^{n+1}. EGF and EGB are EL averaged over the intervals, t^n to t^{n+1} and t^{n-1} to t^n, respectively. It is this maneuver that renders the internal mode immune to the CFL condition based on the barotropic wave speed. The governing wave speed is the baroclinic wave speed.

Structure of the Internal Mode Calculation.

The calculation of the three-dimensional (internal) variables is separated into a vertical diffusion time step and an advection plus horizontal diffusion time step. The

former is implicit (to accommodate small vertical spacing near the surface) whereas the latter is explicit. To illustrate, consider the temperature equation,

$$\frac{\partial DT}{\partial T} + Adv(T) - Dif(T) = \frac{1}{D}\frac{\partial}{\partial \sigma}\left(K_H \frac{\partial T}{\partial \sigma}\right) - \frac{\partial R}{\partial \sigma} \tag{23}$$

$Adv(T)$ and $Dif(T)$ represents the advection and horizontal diffusion terms. The solution is carried out in two steps. Thus, the advection and horizontal diffusion parts are differenced according to

$$\frac{D^{n+1}\tilde{T} - D^{n-1}T^{n-1}}{2\Delta t} = - Adv(T^n) + Dif(T^{n-1}) \tag{24}$$

and is solved by SUBROUTINE ADVT. The vertical diffusion part is differenced according to

$$\frac{D^{n+1}T^{n+1} - D^{n+1}\tilde{T}}{2\Delta t} = \frac{1}{D^{n+1}}\frac{\partial}{\partial \sigma}\left(K_H \frac{\partial T^{n+1}}{\partial \sigma}\right) - \frac{\partial R}{\partial \sigma} \tag{25}$$

and is solved by SUBROUTINE PROFT as detailed in section 9 wherein (25) is first divided by D^{n+1}. Note that, in this subroutine, T^{n-1} is stored in TB, T^n in T and T^{n+1} in UF.

In the "leap frog" time differencing scheme, the solutions at odd time steps can diverge slowly from the solutions at the even time steps. This time splitting is removed by a weak filter (Asselin, 1972) where the solution is smoothed at each time step according to

$$T_s = T + \frac{\alpha}{2}\left(T^{n+1} - 2T^n + T^{n-1}\right)$$

where T_s is the smoothed solution; frequently, we use $\alpha = 0.05$. This technique introduces less damping than either the Euler-backward or forward stepping techniques. After smoothing, T_s is reset to T^{n-1} and T^{n+1} to T^n.

Grid Arrangement

The staggered grid arrangement for the external mode is depicted in Fig. 4 and 5 for the external and internal grid respectively. These diagrams will be useful in understanding the coding in MAIN and in the "PROF" and "ADV" subroutines. The grid can be an orthogonal curvilinear grid. One merely needs to specify h_x(=DX(I, J)) and h_y(=DY(I, J))

as that associated with a particular grid. The advective operators in equations (2) to (8) and (17) to (19) are then described in a finite volume sense; i.e. Equation (5) or, rather, the *Adv* operator in (24), is written

$$-Adv(T)h_x h_y = \delta_x\left(Dh_y UT\right) + \delta_y(Dh_x VT) + h_x h_y \frac{\delta_\sigma(\omega T)}{\delta\sigma} \qquad (26)$$

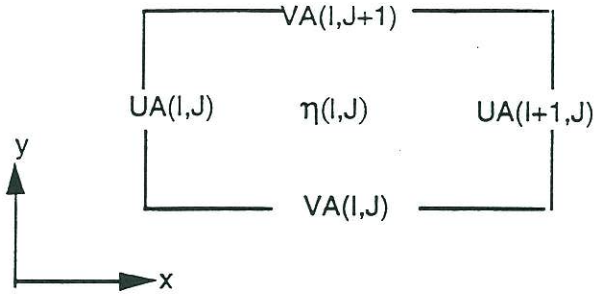


Figure 4. The two-dimensional external mode grid.

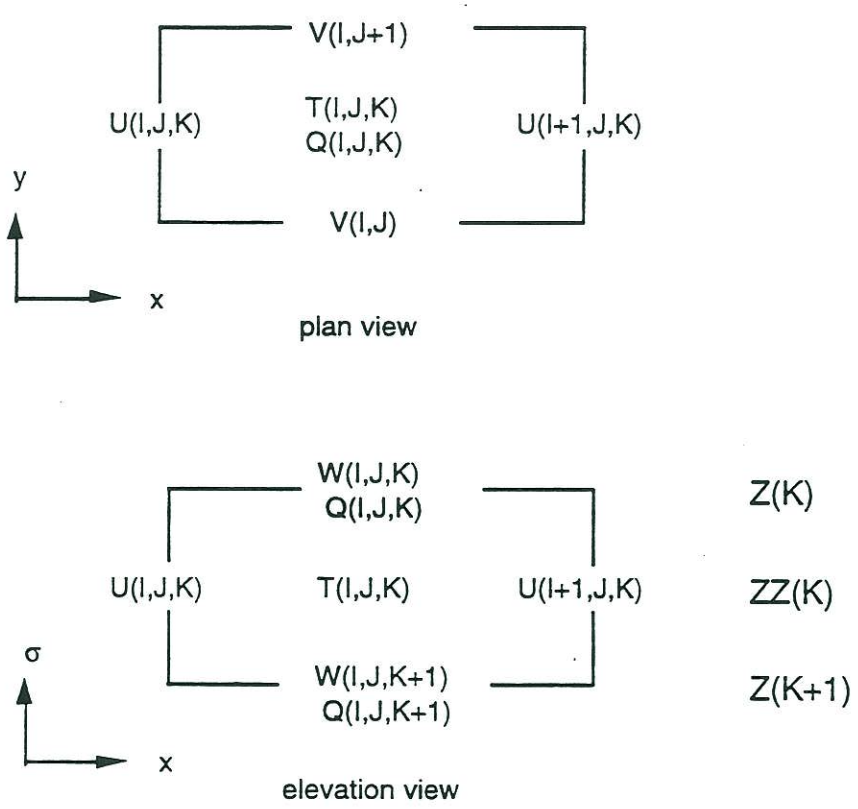


Figure 5. The three-dinensional internal mode grid. Q represents KM, KH, Q2, or Q2L. T represents T,S or RHO.

(where it might be more consistent to multiply through by $\delta\sigma$, but this has not been effected in the code). Thus Dh_yUT represents the transport of T and δ_x represents the difference in this quantity through the opposing faces of the volume element. We leave it to the code listing in SUBROUTINE ADVT to describe the exact method of differencing.

The differencing for the velocity is accomplished in a similar way but involves Coriolis and curvature terms. Thus,

$$-Adv(U)h_xh_y = \delta_x\left(Dh_yU\right) + \delta_y(Dh_xUV) + h_xh_y\frac{\delta\sigma(\omega U)}{\delta\sigma} - \tilde{f}VD\,h_xh_y \qquad (27)$$

where

$$\tilde{f} = \frac{V\delta_x(h_y)}{h_xh_y} + \frac{U\delta_x(h_x)}{h_xh_y} \qquad (28)$$

is the curvature term. In ADVCT, the horizontal advection, diffusion and curvature terms are calculated (and stored in ADVX, ADVY) well in advance of ADVU and ADVV so that their vertical averages can be used in the external mode calculation. In ADVU and ADVV, the pressure gradient, Coriolis and vertical advection are included along with the terms imported from ADVCT.

Time Step Constraints.

The Courant-Friedrichs-Levy (CFL) computational stability condition on the vertically integrated, external mode, transport equations limits the time step according to

$$\Delta t_E \leq \frac{1}{C_t}\left|\frac{1}{\delta x^2} + \frac{1}{\delta y^2}\right|^{-1/2} \qquad (29)$$

where $C_t = 2(gH)^{1/2} + U_{max}$; U_{max} is the expected, maximum velocity expected. There are other restrictions but in practice the CFL limit is the most stringent. The model time step is usually 90% of this limit. The internal mode has a much less stringent time step since the fast moving external mode effects have been removed. The time step criteria is analogous to that for the external mode given by Equation (26) and is

$$\Delta t_I \leq \frac{1}{C_T}\left|\frac{1}{\delta x^2} + \frac{1}{\delta y^2}\right|^{-1/2} \qquad (30)$$

where $C_T = 2C + U_{max}$; C_T is the maximum internal gravity wave speed based on the gravest mode, commonly of order 2m/s, and U_{max} is the maximum advective speed. For

typical coastal ocean conditions the ratio of the time steps, $\Delta t_I / \Delta t_E$ = DTI/DTE, is often a factor of 50 - 80 or larger.

Additional limits are imposed by horizontal diffusion

$$\Delta t_I \leq \frac{1}{4A_H} \left| \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right|^{-1} \tag{31}$$

and rotation

$$\Delta t_I < \frac{1}{f} = \frac{1}{2\Omega \sin \Phi} \tag{32}$$

where A_H is the horizontal diffusivity, Ω is the angular velocity of the earth and Φ is the latitude. (31) and (32) are, however, not restrictive compared to (29) and (30).

5. comblk.h

Common block definitions are contained in the file, comblk.h. The file is then "include"d in each subroutine.

6. PROGRAM MAIN and the external mode

The main program contains model initialization and, subsequently, internal mode time stepping via the index, IINT. All of the internal mode (three-dimensional) subroutines, ADVQ, PROFQ, ADVU, PROFU, ADVV, PROFV, ADVT (for temperature or salinity), PROFT (for temperature or salinity) and DENS are called once for each value of IINT = 1 to IEND.

Imbedded in the IINT loop (which terminates at S.N. 9000) is the external mode IEXT loop which cycles ISPLIT times. Note that DTI/DTE = ISPLIT. The external mode solves for the vertically mean velocities and surface elevation using the density created by the internal mode which is held constant throughout the many external mode, time steps.

The advective and horizontal diffusion terms in the external model are calculated by vertical integration of the corresponding internal terms, ADVX and ADVY, created in subroutine ADVCT. The latter are available every internal time step. However, they are updated by virtue of similar terms (but derived from the mean velocity) in SUBROUTINE ADVAVE. We find that this need not be done every external time step to maintain a stable calculation. SUBROUTINE ADVAVE which calculates these terms are called at intervals of ISPADV; a typical value is ISPADV = 5.

7. SUBROUTINE ADVAVE

This subroutine calculates the advective and horizontal diffusion terms for the external mode calculation contained in equations (18) and (19). If MODE = 2, it also calculates the bottom friction from a quadratic drag equation; otherwise, in the standard three dimensional calculation, the bottom friction is determined by PROFU and PROFV quite naturally as a byproduct of the bottom boundary layer.

8. SUBROUTINE ADVT

This subroutine solves equation (24) for temperature or salinity (or any other scalar variable) which are labeled F internally. $\widetilde{D}$ has been set to D^{n+1}. The operator $Adv(F)$ is written in the form of equation (26). As shown in the code listing, horizontal advective transports through the faces of the grid elements are computed in the form, D*U*F*DY and D*V*F*DX, using appropriate cell averages. Note that, in the code listing, DT is simply the external value, D, averaged over the internal time step. To the advective fluxes are added the horizontal diffusion fluxes. Before this occurs, TCLIM is subtracted from the actual temperature [see Mellor and Blumberg, 1986, and discussion after equation (12a, b)]. Then, the diffusion terms will (gently) drive the calculated field back to climatology. As resolution improves, the diffusion terms decrease as DX DY decreases. The vertical advective flux divergence is determined (and temporally stored in FF) and then combined with the horizontal transport divergence. Finally, the time step is executed and the new value is stored in FF.

9. SUBROUTINE PROFT

This subroutine solves (25) for temperature and salinity. We use the method described on p.198-201 of Richtmeyer and Morton (1967). The procedure here will be a model for U,V,Q2 and Q2L in which case the radiation term in (25) is either null or is replaced by source/sink terms. Subroutine PROFT as well as ADVT can be used to solve for other geochemical constituents besides temperature and salinity.

First, finite difference (25) with respect to σ. [We note that $\widetilde{D} = D^{n+1}$; the choice is irrelevant so long as the same value of $\widetilde{D}$ is used in (24) and (25)]. Thus, with reference to the elevation view of Fig. 5.

$$F_k - \tilde{F}_k = \frac{DTI2}{D**2**DZ_k}\left[\frac{KH_k}{DZZ_{k-1}} * (F_{k-1} - F_k) - \frac{KH_{k+1}}{DZZ_k} * (F_k - F_{k+1})\right] - \frac{DTI2}{D * DZ_k}\left[RAD_k - RAD_{k+1}\right] \quad (9\text{-}1)$$

where $DZ_k = Z_k - Z_{k+1}$, $DZZ_k = ZZ_k - ZZ_{k+1}$ and F_k represents either temperature or salinity. In the above, we use subscripts for k instead of parenthetical enclosure to save space; we also omit the I, J indicies.

Solution Technique

Equation (9-1) can now be written as

$$-F_{k+1} * A_k + F_k * (A_k + C_k - 1) - F_{k-1} * C_k = D_k \quad (9\text{-}2)$$

where

$$A_k = - \frac{DTI2 * KH_{k+1}}{D**2 * DZ_k * DZZ_k} \quad (9\text{-}3a)$$

$$C_k = - \frac{DTI2 * KH_k}{D**2 * DZ_k * DZZ_{k-1}} \quad (9\text{-}3b)$$

$$D_k = - \tilde{F}_k + \frac{DTI2}{D * DZ_k}\left[RAD_k - RAD_{k+1}\right] \quad (9\text{-}3c)$$

Now assume a solution of the form

$$F_k = E_k * F_{k+1} + G_k \quad (9\text{-}4)$$

Inserting F_{k+1} directly from (9.4) and F_{k-1} ,obtained from (9.4), into (9.2) and collecting coefficients of F_k and 1 yields

$$E_k = \frac{A_k}{A_k + C_k * (1 - E_{k-1}) - 1} \quad (9\text{-}5a)$$

$$G_k = \frac{C_k * G_{k-1} - D_k}{A_k + C_k * (1 - E_{k-1}) - 1} \quad (9\text{-}5b)$$

The way the system works is as follows: All A_k's, C_k's and D_k's are calculated from (9-3a,b,c). Surface boundary conditions, discussed below, provide E_1 and G_1 and all of the

necessary E_k's and G_k's are obtained from the descending (as k increases towards the bottom) recursive relations (9-5a,b). Bottom boundary conditions provide F_{kb-1} where kb-1 is the grid point nearest the bottom. Thereafter all of the F_k's may be obtained from the ascending recursive relation (9-4)

Short Wave Radiation

To specify the short wave radiation, we use the classification of Jerlov(1976), but approximate his methodology such that short wave radition, absorbed in the first several meters, is added to the surface boundary condition. Thus, we will shortly install WTSURF + (1.–TR)*SWRAD as the surface boundary condition so that the remainder is attenuated according to

$$RAD_k = SWRAD * TR * EXP(EXTINC * Z_k * D) \qquad (9\text{-}6)$$

A very good fit to Jerlov's tabulated attenuation function is given by (9-8) if we set the constants TR and EXTINC according to the following table:

NTP	TYPE	TR	EXTINC(m^{-1})
1	I	.32	.037
2	IA	.31	.042
3	IB	.29	.056
4	II	.26	.073
4	III	.24	.127

Surface and Bottom Boundary Conditions

To apply the surface boundary conditions where the surface flux is prescribed (prescribing the surface temperature is much simpler such that $E_1 = 0$, $G_1 = F_1$) begins with (9-1) where for k=1

$$F_1 - \tilde{F}_1 = \frac{DTI2}{D * DZ_1}(-WTSURF - (1.-TR) * SWRAD) + A_1(F_1 - F_2)$$

Using (9-4) to eliminate F_2 and collecting coefficients of F_1 and 1 yields

$$E_1 = \frac{A_1}{A_1 - 1} \qquad (9\text{-}7a)$$

$$G_1 = \left[\frac{DTI2 * (WTSURF + (1.-TR) * SWRAD)}{D * DZ_1} - \tilde{F}_1\right] * \left[\frac{1}{A_1 - 1}\right] \qquad (9\text{-}7b)$$

At the bottom, we specify zero heat flux. A repeat of the above procedure leads to

$$F_{kb-1} = \frac{C_{kb-1} * G_{kb-2} - \tilde{F}_{kb-1}}{C_{kb-1} * (1 - E_{kb-1}) - 1} \qquad (9\text{-}8)$$

10. SUBROUTINE BAROPG

This subroutine calculates the baroclinic, vertical integrals involving density in equation (3) and (4) after the equations have been written in finite volume form.

We note the fact that, in the code, $\overline{\rho}$ = RMEAN has been subtracted from ρ before the integrals are calculated. $\overline{\rho}$ is the basin area average density which is mapped onto the sigma-grid just as the initial conditions were similarly mapped This procedure removes most of the truncation error in the transformed baroclinic terms which arise due to the subtraction of the two large terms involving $\partial\rho / \partial x$ and $D^{-1}(\partial D/\partial x)\sigma\partial\rho / \partial\sigma$ in (3) and similarly in (4)

11. SUBROUTINES ADVCT, ADVU AND ADVV

ADVCT calculates the horizontal advection (including curvature terms) and the diffusion parts of (3) and (4) which are differenced in the manner of equation (27) and saved as ADVX and ADVY. These terms are vertically integrated and saved as ADX2D and ADY2D for use in the external mode calculation in program MAIN. Originally, ADVCT had been incorporated in ADVU and ADVV. However, it was determined by Oregon State University colleagues that advancing the calculation of horizontal advection terms (see Figure 2) for use in the external mode increased the models intrinsic stabillity.

12. SUBROUTINES PROFU AND PROFV

These subroutines are virtually identical to SUBROUTINE PROFT. However, the bottom boundary conditions are obtained from equations (14c,d,e).

13. SUBROUTINE ADVQ

This subroutine is very similar to all the other "ADV-" subroutines in that it calculates the advective terms for the the turbulence quantities, Q2 and Q2L.

14. SUBROUTINE PROFQ

This subroutine first solves for the vertical part of the equations (7) and (8) for Q2 and Q2L in the manner of equation (25). The numerical procedure is the same as SUBROUTINE PROFT. The turbulence closure scheme as described by Mellor and Yamada(1982) is contained in this subroutine. A somewhat simplified version of the level 2 1/2 model is used here and is discussed in Galperin *et al* (1988) and Mellor (1989).

The vertical diffusivities, K_M and K_H, are defined according to

$$K_M = q\ell\, S_M \tag{14-1a}$$

$$K_H = q\ell\, S_H \tag{14-1b}$$

The coefficients, S_M and S_H, are functions of a Richardson number given by

$$S_H[1-(3A_2B_2 + 18A_1A_2)G_H] = A_2[1-6A_1 / B_1] \tag{14-2a}$$

$$S_M[1-9A_1A_2G_H] - S_H[(18A_1 2 + 9A_1A_2)G_H] = A_1[1 - 3C_1 - 6A_1 / B_1] \tag{14-2b}$$

where

$$G_H = -\frac{\ell^2}{q^2}\frac{g}{\rho_o}\left[\frac{\partial \rho}{\partial z} - \frac{1}{c_s^2}\frac{\partial p}{\partial z}\right] \tag{14-2c}$$

is a Richardson number. The five constants in (14-2a,b) are mostly evaluated from near surface turbulence data (law-of-the-wall region) and are found (Mellor and Yamada, 1982) to be (A_1, A_2, B_1, B_2, C_1) = (0.92, 16.6, 0.74, 10.1, 0.08). The stability functions limit to infinity as G_H approaches the value, 0.0288, a value larger than one expects to find in nature.The quantity, c_s^2, in the square brackets of (14-2c) is the speed of sound squared. In the code the vertical pressure gradient is obtained from the hydrostatic relation,of course, but here, the density is taken as a constant consistent with the pressure determination in SUBROUTINE DENS; i.e., $\partial p / \partial z = -\rho_o g$

15. SUBROUTINE VERTVL

This short subroutine integrates equation (2) to obtain the sigma coordinate transformed "vertical velocity" which, actually, is the velocity normal to sigma surfaces. Occasionally, check W(I, J, KB); if all is well, the code should yield very small values ($\sim 10^{-11}$).

16. SUBROUTINE BCOND

Lateral boundary conditions contiguous to coastlines are handled automatically by the masks DUM, DVM and FSM. They set to zero the velocities normal to land boundaries. The landward tangential velocities in the horizontal friction terms are also set to zero. For a sigma coordinate system, the latter is of little importance since the minimum water depth next to the coast can be quite shallow so that bottom friction dominates over lateral friction. We generally set the minimum depth in the range, 10 to 20 m, but a numerically acceptable minimum depth has not been firmly established.

Open boundaries are considerably more demanding and uncertain and there is a need for boundary conditiion specification for both the external and internal modes.

External Mode

We illustrate discussion with reference to an "eastern" boundary condition. Common choices for normal, vertically mean velocities or elevation are listed in Table A. In the code, the boundary variables, UABE(J) or ELE(J) may be used to convey the variable, BC(J).

In addition to the *primary boundary conditions* in Table A, one needs to accomodate the calculation of momentum fluxes. Thus, when U(IM,J) is specified, also set ELF(IM,J)=ELF(IMM1,J); when ELF(IMM1,J) is specified, also set UAF(IM,J)=UAF(IMM1,J) and ELF(IM,J)=ELF(IMM1,J).

Calculations do not seem overly sensitive to the velocity component tangential to the boundary, at least for low Rossby number flows. We often set it to zero; alternatively an advective boundary condition such as (16-6) has been used.

Internal Mode

In the equations for scalars, boundary conditions are needed only for the non-linear, advective parts of these equations. On boundary points, an upstream advection equation is used. Thus, in the case of temperature at a boundary, I = IM, we use the following equation:

$$TF(IM) = T(IM) + (DTI / DX) * [U1 * (T(IM) - T(IM-1)) + U2 * (TBE - T(IM))] \quad (16\text{-}6)$$

where U1 = 0.5*(U(IM) + ABS(U(IM))) and U2 = 0.5*(U(IM) - ABS(U(IM)); when U is negative, U1 = 0; when U is positive U2 = 0. The quantity TBE is an "eastern"

TABLE A: A list of possible, external mode, "eastern" boundary conditions. For the radiation boundary conditions, change the sign before $c_e \equiv \sqrt{gH}$ when applied to a "western" boundary. Also, for a western boundary condition, IM $\rightarrow$ 2 and IMM1 $\rightarrow$ 3. The variable, BC, is user specified and may be equated to the left sides of (16.1) - (16.3) where $\overline{U}$ and η are known *a priori*. The right sides of (16-4) and (16-5) need not necessarily be zero.

Formula		CODE
$D\overline{U} = BC$	(16-1)	UAF(IM,J)=0.5*BC(J)/(H(IM,J)+ELF(IM,J) + H(IMM1,J)+ELF(IMM1,J))
$\eta = BC$	(16-2)	ELF(IMM1,J)=BC(J)
$H\overline{U} - c_e\eta = BC$	(16-3)	UAF(IM,J)=SQRT(GRAV/(H(IMM1,J)) *EL(IMM1)+BC(J)
$\partial\overline{U}/\partial t + c_e\,\partial\overline{U}/\partial x = 0$	(16-4)	GAE=DTE*SQRT(GRAV*H(IM,J)) /DX(IM,J) UAF(IM,J)=GAE*UA(IMM1,J) +(1.– GAE)*UA(IM,J)
$\partial\eta/\partial t + c_e\,\partial\eta/\partial x = 0$	(16-5)	GAE=DTE*SQRT(GRAV*H(IMM1,J)) /DX(IMM1,J) ELF(IMM1,J)=GAE*EL(IMM2,J) +(1.– GAE)*EL(IMM1,J)
CYCLIC	(16-6)	ELF(1,J)=ELF(IMM2,J) ELF(2,J)=ELF(IMM1,J) ELF(IM,J)=ELF(3,J) UAF(2,J)=UAF(IMM1,J) UAF(IM,J)=UAF(3,J) VAF(2,J)=VAF(IMM1,J) VAF(IM,J)=VAF(3,J)

boundary value of the temperature and must be supplied by the user. Salinity is handled in identical fashion. It will be seen that these boundary data are only used when the flow is into the domain. (In the case of tangential velocity, the inflow "data" is generally taken to be zero). This boundary condition for scalar quantities seems to work well.

As in the external model, the choice for the normal velocities is unclear. One might

TABLE B: A list of possible internal mode, normal velocity, boundary conditions. For the radiation boundary condition, change the sign before c_i when applied to a "western" boundary. The variable, BC, is user specified.

Formula		CODE
$U = BC$	(16-7)	UF(IM,J,K)=BC(J,K)
$\partial U/\partial t + c_i\,\partial U/\partial x = 0$	(16-8)	GAI=SQRT(H(IM,J)/HMAX) UF(IM,J,K)=GAI*U(IMM1,J,K) +(1.– GAI)*U(IM,J,K)
CYCLIC	(16-9)	After adding K's; for velocity, UF and VF same as UAF and VAF; for tenperature and salinity, UF and VF same as ELF

presume that (16-8) is to be preferred over (16-7) since internal waves can pass through the boundary with little reflection; in some applications, that may be the case. However, we have seen cases (open boundaries with substantial inflows) where the "freedom" of (16-8) in conjunction with (16-6) can set up unphysical, but numerically valid, baroclinic structures interior to the boundary.

The finite difference expression one gets from (16-8) is

$$U_{im}^{n+1} = \gamma U_{im-1}^{n} + (1-\gamma)U_{im}^{n}; \quad \gamma \equiv c_i \Delta t_i / \Delta x$$

where one might like c_i to be the gravest mode, baroclinic phase speed. However, it is assumed that: *a)* the user has found and is using a Δt_i such the maximum value of γ is near unity, corresponding approximately to the maximum depth and *b)* that c_i is proportional to $\sqrt{H}$. This is a seemingly crude approximation, but may perform fairly well; it at least guarantees that $0 < \gamma \leq 1$.

17. SUBROUTINE DENS

The UNESCO equation of state, as adapted by Mellor(1991) is used. The *in situ* density is determined as a function of salinity, potential temperature and pressure; the latter is approximated by the hydrostatic relation and constant density. Initially, the values TBIAS and SBIAS are subtracted from temperature and salinity to reduce round-off error. With 32 bit arithmatic, a suggestion is TBIAS = 10. and SBIAS = 35. for open ocean

models; with 64 bits, zero values are appropriate. In DENS, these values are added again before the density is calculated The actual density is normalized on 1000 kg/m^2. Since only gradients are needed (in subroutines BAROPG and PROFQ), the value 1.025 is subtracted to reduce round-off error. APPENDIX A includes some discussion of thermodynamics.

18. SUBROUTINE SLPMIN

This subroutine examines the topography and adjusts H(I,J) so that the difference of the depths of any two adjacent cells divided by the sum of the depths is less than or equal to the parameter, SLMIN. In the process, volume is preserved. What generally happens is that the topography in deeper water is not changed whereas the shallower regions are altered depending on resolution.

19. utililty subroutines

There are a number of utility subroutines supplied with the program. For the most part they can be understood by reference to comments written into the code. All of the printing subroutines except SUBROUTINE EPRXYZ print out numbers in integer format. They accept a scale factor in the argument list which is either zero, in which case the code generates its own scale factor, or a finite value which is then used to scale the printed numbers.

20. PROGRAM CURVIGRID

The program is set up to accept values of longitude and latitude, here denoted by x and y to define the four edges of the gridded domain. This can be altered to accommodate rectilinear coordinates by setting the cosine of the latitude, CS = 1, in subroutine ORTHOG or by expunging the variable
completely.

The border of the domain is determined by NB, NR or NL points on the J=1, I=1 and J=JM borders respectively. In this version of the program, DATA statements contain this information. Cubic splines are then used to fill in the missing border coordinates.

The program is comprised of two steps:

I. The interior grid points (1 < J < JM) are then filled such that the values at every I column is distributed proportionately to the y-values at I=1; the interior x value are similarly distributed.

II. Subroutine ORTHOG is called to render the $x_{i,j}$ and $y_{i,j}$ an orthogonal coordinate system. Then, use is made of the orthogonality conditions

$$\left(\frac{\partial x}{\partial s}\right)_j = -\left(\frac{\partial y}{\partial s}\right)_i, \quad \left(\frac{\partial y}{\partial s}\right)_j = \left(\frac{\partial x}{\partial s}\right)_i \tag{19-1a,b}$$

or

$$\frac{\delta_j x}{\delta_j s} = -\frac{\delta_i y}{\delta_i s}, \quad \frac{\delta_j y}{\delta_j s} = \frac{\delta_i x}{\delta_i s} \tag{19-2a,b}$$

With reference to Fig. 6, (2a,b) are solved according to

$$x_{i,j} - x_{i,j-1} = \frac{\delta_j s}{\delta_i s}[y_{i+1,j} - y_{i-1,j} + y_{i+1,j-1} - y_{1,j-1}] \tag{19-3a}$$

$$y_{i,j} - y_{i,j-1} = \frac{\delta_j s}{\delta_i s}[x_{i+1,j} - x_{i-1,j} + x_{i+1,j-1} - x_{1,j-1}] \tag{19-3b}$$

where

$$\delta_i s = \frac{1}{4}[(x_{i+i,j} - x_{i,j})^2 + (y_{i+1,j} - y_{i1,j})^2]^{1/2} + \frac{1}{4}[(_{i+1,j-1} - x_{i-1,j-1})^2 + (y_{i+1,j-1} - y_{i-1,j-1})^2]^{1/2} \tag{19-4a}$$

$$\delta_j s = [(x_{i,j} - x_{i,j-1})^2 + (y_{i,j} - y_{i,j-1})^2]^{1/2} \tag{19-4b}$$

The factor, CS, the cosine of the latitude, is not included in (19-3a,b) and (19-4a,b) but is included in the corresponding equations in ORTHOG. Now, the above equations are interated many times during which $\delta_j s$ is fixed; i.e., $\delta_j s$, $x_{i,1}$ and $y_{i,1}$ are data of the initial field specified in step I which are retained. In the course of iteration, $\delta_j s$, $x_{i,j}$ and $y_{i,j}$ are reevaluated. The shape of the original domain does change but not greatly. During this iteration, CS is held fixed. In fact, CS changes very little so that ORTHOG is called only twice to converge on this factor.

It should be noted that, if the border points contain too much curvature, then the curves normal to the i = constant curves can focus to a point at some j row after which the calculation is nonsense. Some trial and error is therefore required. A way to avoid this is to call POISSON after step I which solves for $y_{i,j}$ according to $\partial^2 y / \partial^2 i + \partial^2 y / \partial^2 j = 0$. This avoids the focusing problem but may not yield the most desireable grid.

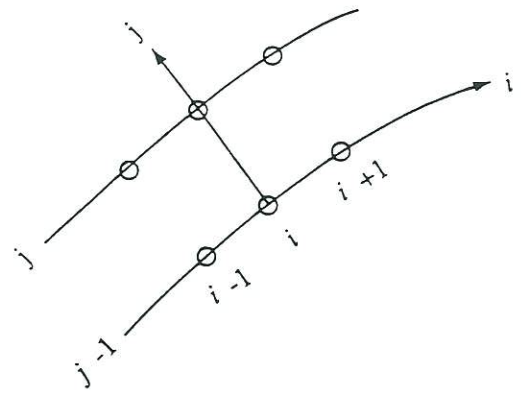


Figure 6. The orthogonal curvilinear grid system.

A good practice is to map the bottom topography on to the grid, then calculate the CFL limiting time step for each grid point; one wishes, of course, to avoid overly small steps.

APPENDIX A: Note on the Equation of State, Potential Temperature and Static Stability

Two equations of state for density are

$$\rho = \rho_1(T,S,p) \tag{A-1}$$

$$\rho = \rho_2(\Theta,S,p) \tag{A-2}$$

where T is *in situ* temperature and Θ is the potential temperature. In the model, (A-2) is used. To relate potential temperature, Θ, to *in situ* temperature, T, recall the thermodynamic relation for entropy.

$$Td\eta = dh - \frac{dp}{\rho} - \mu dS \tag{A-3}$$

where η is the entropy, h, enthalpy and μ the chemical potential for salt taken here as a single average constituent. Furthermore,

$$dh = C_p dT + (1 - \alpha T)\frac{dp}{\rho} \tag{A-4}$$

where we have set

$$\left(\frac{\partial h}{\partial T}\right)_{p,S} \equiv C_p \ , \quad \left(\frac{\partial h}{\partial p}\right)_{T,S} = \frac{(1-\alpha T)}{\rho} \qquad \text{(A-5a, b)}$$

and where the coefficient of thermal expansion is

$$\alpha \equiv -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_p \qquad \text{(A-5c)}$$

We note that (A-5b) has been obtained from (A-3) and one of Maxwell's relations. Combining (A-3) and (A-4), we have

$$d\eta = C_p \frac{dT}{T} - \alpha \frac{dp}{\rho} - \frac{\mu dS}{T} \qquad \text{(A-6)}$$

The definition of potential temperature in oceanography* is

$$C_{po} \frac{d\Theta}{\Theta} \equiv C_p \frac{dT}{T} - \alpha \frac{dp}{\rho} \qquad \text{(A-7)}$$

where $C_p = C_p\,(T,S,p)$ and $C_{po} = C_p\,(T,S,0)$. Combining (A-6) and (A-7),

$$d\eta = C_{po} \frac{d\Theta}{\Theta} - \frac{\mu dS}{T} \qquad \text{(A-8)}$$

For processes where heat transfer, viscous dissipation and salt diffusion are null, $D\Theta/Dt = DS/DT = 0$; then, from (A-8), $D\eta/Dt = 0$; i.e. the process is isentropic. An integral relation obtained from (A-7) is

$$T(z) - \Theta = \int_p^o \frac{\alpha' T'}{C_p'} \frac{dp'}{\rho'} = -\int_z^o \frac{\alpha' T' g}{C_p'} dz' \qquad \text{(A-9)}$$

The hydrostatic pressure relation is used to obtain the sencond expression on the right of the equal sign. In (A-9), $\Theta(z) = \Theta(0) = T(0)$. For $T = 10°C$, $S = 35$ psu and $p = 0$, one finds (Gill, 1982, p.603) that $\alpha T g / C_p \cong 0.12K / 1000m$. Equation (A-9) allows one to

---

* as contrasted to meteorology, where, for a perfect gas, we have $\alpha T = 1$ and $p = \rho RT$. Potential temperature is then defined as $d\Theta/\Theta = d\eta/C_p = dT/T - (R/C_p)dp/p$ which can be integrated exactly to give $\Theta = T(p_o/p)^{R/C_p}$; p_o is a reference pressure where $\Theta = T$.

initialize potential temperature in the model given *in situ* properties. An algorithm to do this is provided by Bryden (1973).

Static Stability

To conveniently provide further background information and also inquire into an aspect of the Boussinesq approximation, we review the following equations for two-dimensional isentropic flow.

$$\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{w}}{\partial z} = 0 \tag{A-10}$$

$$\tilde{\rho}\left(\frac{\partial \tilde{u}}{\partial t} + \tilde{\mathbf{u}} \cdot \nabla \tilde{u}\right) = - \frac{\partial \tilde{p}}{\partial z} \tag{A-11}$$

$$\tilde{\rho}\left(\frac{\partial \tilde{w}}{\partial t} + \tilde{\mathbf{u}} \cdot \nabla \tilde{w}\right) = - \frac{\partial \tilde{p}}{\partial z} - \tilde{\rho} g \tag{A-12}$$

where we have made the Boussinesq approximation in (A-10) but have not done so in (A-11) and (A-12). Equation (A-10) is justified by examination of the full equation $\nabla \cdot \mathbf{u} + \tilde{\rho}^{-1} D\tilde{\rho} / Dt = 0$. The first term scales like u_o/L whereas the second scales as $(u_o/L)\, \delta\tilde{\rho} / \tilde{\rho}$ Since $\delta\tilde{\rho} / \tilde{\rho} \lesssim .05$ in the ocean, the second term can be neglected.

Let mean quantities be denoted by upper case letters and fluctuating quantities by lower case letters; the exception to this is density where ρ and ρ' are the mean and fluctuating values. For this analysis the mean velocity will be zero. Therefore we have $(\tilde{u}, \tilde{w}) = (u, w)$, $\tilde{p} = P + p$, $\tilde{\rho} = \rho + \rho'$, $\partial P / \partial z = -\rho g$ and $\rho = \rho(z)$ so that, for small perturbations,

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \tag{A-13}$$

$$\rho \frac{\partial u}{\partial t} = - \frac{\partial p}{\partial z} \tag{A-14}$$

$$\rho \frac{\partial u}{\partial t} = - \frac{\partial p}{\partial z} - \rho' g \tag{A-15}$$

Now for isentropic flows the equations of state yields $D\tilde{\rho} / Dt = c^{-2} D\tilde{p} / Dt$ where $c^2 \equiv (\partial\tilde{p} / \partial\tilde{\rho})_{\Theta,S}$ is the speed of sound squared. The corresponding density perturbation equation is

$$\frac{\partial \rho'}{\partial t} + w\frac{\partial \rho}{\partial z} = \frac{1}{c^2}\left(w\frac{\partial p}{\partial z} + \frac{\partial p}{\partial t}\right)$$

or

$$\frac{\partial \rho'}{\partial t} - w\frac{\rho N^2}{g} = \frac{1}{c^2}\frac{\partial p}{\partial t} \qquad \text{(A-16)}$$

where

$$\rho\frac{N^2}{g} \equiv -\frac{\partial \rho}{\partial z} + \frac{1}{c^2}\frac{\partial p}{\partial z} = -\frac{\partial \rho}{\partial z} - \frac{\rho g}{c^2} \qquad \text{(A-17)}$$

N^2 is the Brunt-Vassala frequency squared or the static stability. If one eliminates u, p and ρ' from (A-13) to (A-16), the resulting equation for w is

$$\frac{\partial^2}{\partial t^2}\left[\frac{\partial^2 w}{\partial z^2} + \frac{\partial^2 w}{\partial x^2} + \frac{N^2}{g}\frac{\partial w}{\partial z}\right] + N^2\frac{\partial^2 w}{\partial x^2} = 0 \qquad \text{(A-18)}$$

The last term in the square brackets can be neglected compared with the first. Thus, $g^{-1}N^2 w_z / w_{zz} \simeq g^{-1}N^2 L_z$ where L_z is the vertical scale height. $g^{-1}N^2 L_z$ has two parts as shown in (A-17). If we take $L_z \approx 1000$m, then the first part, $|\rho^{-1}\rho_z L_z| \approx .010$ and the second part, $c^{-2}gL_z \approx .005$. Tracing back through the original equations, we find that this approximation is equivalent to setting ρ = constant = ρ_o in (A-14) and (A-15) and neglecting the right side of (A-16).

A solution to (A-18) for N^2 = constant is $w \propto \exp[i(lz + kx - \sigma t)]$ where the dispersion relation is $\sigma^2 = N^2k^2 / (l^2 + k^2)$ If $N^2 < 0$, the flow is unstable; if $N^2 > 0$, the flow is stable. Thus, N^2, given by (A-17) is the correct static stability parameter for use in the turbulence closure model which are constructed from perturbation equations like (A-16) together with other eqautions and terms.

References

Andre, J. C., G. DeMoor, G. Therry, and R. DuVachat, Modeling the 24-hour evolution of the mean and turbulence structures of the planetary boundary layer, J. Atmos. Sci., 35, 1861-1863, 1978.

Asselin, R., Frequency filters for time integrations, Mon. Weather Rev., 100, 487-490, 1972.

Baringer, M. O., and J. F. Price, Mixing and spreading of the Mediterranean outflow, J. Phys. Oceanogr., submitted, 1996.

Blumberg, A. F.,and G. L. Mellor, A coastal ocean numerical model, in Mathematical Modelling of Estuarine Physics, Proc. Int. Symp., Hamburg, Aug. 1978, edited by J. Sunderman and K.-P. Holtz, pp.203-214, Springer-Verlag, Berlin, 1980.

Blumberg, A.F., and G.L. Mellor, Diagnostic and prognostic numerical circulation studies of the South Atlantic Bight, J. Geophys. Res., 88, 4579-4592, 1983.

Blumberg, A.F., and G.L. Mellor, A description of a three-dimensional coastal ocean circulation model, in Three-Dimensional Coastal Ocean Models, Vol. 4, edited by N.Heaps, pp. 208, American Geophysical Union, Washington, D.C., 1987.

Bryden, H. L., New polynomials for thermal expansion, adiabatic temperature gradient, and potential temperature of sea water, Deep-Sea Res., 20, 401-408, 1973.

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

Gill, A.E., Atmosphere-Ocean Dynamics, 662 pp., Academic Press, New York, 1982.

Jerlov, N. G., Marine Optics, 14, 231 pp., Elsvier Sci. Pub. Co., Amsterdam, 1976.

Jungclaus, H., and G. L. Mellor, A three-dimensional model study of the Mediterranean out flow, J. Mar. Systems, submitted, 1996.

Klein, P., A simulation of the effects of air-sea transfer variability on the structure of the marine upper layers, J. Phys. Oceanogr., 10, 1824-1841, 1980.

Knudsen, M., Hydrographical Tables. G.E.C. Gad, Copenhagen, pp., Williams and Norgate, London, 19010.

Madala, R. V., and S. A. Piacsek, A semi-implicit numerical model for baroclinic oceans, J. Comput. Phys., 23, 167-178, 1977.

Martin, P.J., Simulation of the mixed layer at OWS November and Papa with several models, J. Geophys. Res., 90, 903-916, 1985.

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

Mellor, G. L., Retrospect on oceanic boundary layer modeling an second moment closure, Hawaiian Winter Workshop on "Parameterization of Small-Scale Processes", January 1989, University of Hawaii, Honolulu, Hawaii, 1989.

Mellor, G.L., Analytic prediction of the properties of stratified planetary surface layers., J. Atmos. Sci., 30, 1061-1069, 1973.

Mellor, G.L., and A.F. Blumberg, Modeling vertical and horizontal diffusivities with the sigma coordinate system, Mon. Wea. Rev, 113, 1380-1383, 1985.

Mellor, G.L., and T. Yamada, A hierarchy of turbulence closure models for planetary boundary layers, J. Atmos. Sci., 31, 1791-1806, 1974.

Mellor, G. L., L. H. Kantha, and H. J. Herring, On Gulf Stream frontal eddies. A numerical experiment, Ocean Modelling, 68, 7-11, 1986.

Mellor, G.L., An equation of state for numerical models of oceans and esturaries. J. Atmos. Oceanic Tech. 8, 609-611, 1991.

Mellor, G. L., and X. H. Wang, Pressure compensation and the bottom boundary layer, J. Phys. Oceanogr., in press, 1996.

Oey, L.-Y., G.L. Mellor, and R.I. Hires, A three-dimensional simulation of the Hudson-Raritan estuary. Part I: Description of the model and model simulations, J. Phys. Oceanogr., 15, 1676-1692, 1985a.

Oey, L.-Y., G.L. Mellor, and R.I. Hires, A three-dimensional simulation of the Hudson-Raritan estuary. Part II: Comparison with observation, J. Phys. Oceanogr., 15, 1693-1709, 1985b.

Oey, L.-Y., G.L. Mellor, and R.I. Hires, A three-dimensional simulation of the Hudson-Raritan estuary. Part III: Salt flux analyses, J. Phys. Oceanogr., 15, 1711-1720, 1985c.

Phillips, N. A., A coordinate system having some special advantages for numerical forecasting, J. Meteorol., 14, 184-185, 1957.

Richtmyer, R. D., and K. W. Morton, Difference Methods for Initial-Value Problems, 2nd Ed., pp., Interscience, New York, 1967.

UNESCO. Tenth rep. of the joint panel on oceanographic tables and standards. UNESCO Tech. Pap.in Marine Science No. 36. UNESCO, paris, 25pp., 1981

Simons, T. J., Verification of numerical models of Lake Ontario. Part I, circulation in spring and early summer, J. Phys. Oceanogr., 4, 507-523, 1974.

Zavatarelli, M., and G. L. Mellor, A numerical study of the Mediterranean Sea Circulation, J. Phys. Oceanogr., 25, 1384-1414, 1995.

Institute for
Naval Oceanography

SP-5
MARCH 1991

A USER'S MANUAL FOR THE PRINCETON NUMERICAL OCEAN MODEL

WILLIAM P. O'CONNOR
INSTITUTE FOR NAVAL OCEANOGRAPHY

The Institute for Naval Oceanography (INO) is operated by the University Corporation for Atmospheric Research (UCAR) under sponsorship of the Naval Oceanographic and Atmospheric Research Laboratory (NOARL). Any opinions, findings, and conclusions or recommendation expressed in this publication are those of the author(s) and do not necessarily reflect the views of NOARL.

ACKNOWLEDGMENTS

This report aids in documenting the primitive equation numerical ocean model originally developed at Princeton University and Dynalysis of Princeton by Drs. George L. Mellor, Alan F. Blumberg, Lakshmi H. Kantha, H. James Herring, Lie-Yauw Oey, Boris Galperin, and others. The model code has evolved slightly as it has been used by different institutions for their purposes and optimized for their computers. This guide documents the version currently in use at the Institute for Naval Oceanography (INO), and may differ slightly from other versions. Thanks are due to Dr. George L. Mellor and especially to Dr. Lakshmi H. Kantha for their explanations of the model. The proofreading was done by Ms. Lydia Harper and Ms. Evelyn Lott.

CONTENTS

1. INTRODUCTION

This is a user's guide for the numerical ocean model developed at Princeton University and presently in use at the Institute for Naval Oceanography. The basic model has been applied successfully to study such diverse regions as Chesapeake Bay (Blumberg 1977), the South Atlantic Bight (Blumberg and Mellor 1983), the Mid Atlantic Bight (Blumberg and Kantha 1985), the Gulf of Mexico (Blumberg and Mellor 1985), New York Harbor (Oey et al 1985a,b,c), Delaware Bay (Galperin and Mellor 1990a,b), and the western Atlantic including the Gulf Stream (Mellor and Ezer 1991). It can model regions from the size of bays and estuaries, to basin scale North Atlantic domains. A good discussion of modeling bays and coastal oceans is given by Blumberg and Oey (1985).

The model is quite developed in its representation of physical processes. The main model characteristics are as follows:

- Primitive equations
- Fully three–dimensional
- Nonlinear
- Flux form of equations
- Boussinesq and hydrostatic
- Terrain following vertical coordinate (σ–coordinate)
- Generalize orthogonal coordinates
- Baroclinic mode
- Free upper surface with barotropic mode
- Turbulence model for vertical mixing
- Smagorinsky horizontal diffusion
- Leapfrog (centered in space and time) time step
- Implicit time scheme for vertical mixing
- Arakawa–C staggard grid

The model is described in several references. The most complete description of the model equations and finite differencing is given by Blumberg and Mellor (1987). A description of the diffusivities in the σ–coordinate system is given by Mellor and Blumberg (1985). The description of the present model cast in generalized orthogonal curvilinear coordinates is given by Blumberg and Herring (1987). A documentation for the use of the model has been produced by Mellor (1990). This user's guide will not attempt to duplicate the above model

descriptions. Rather, it will compliment them by concentrating on how to set parameters in the code for specific applications. The user should first become familiar with the above references, and then use this guide as a supplement when applying the model to a given problem.

The model is configured as a MAIN program and about a dozen subroutines. Generally, only the MAIN program and subroutine BCOND containing the boundary conditions, need to be changed when setting up the model for a specific application. The remainder of the subroutines perform physical calculations common to all applications of the model, such as calculating density and horizontal and vertical advections. This guide will discuss the symbols, arrays, and constants in the MAIN program. The code in the MAIN program is then described, along with the application of the boundary conditions in subroutine BCOND. The use and application of the utilities subroutines to print out arrays of data for easy visual inspection are explained. Then follows a detailed discussion of the necessary input that the user must specify to initialize the model for a specific application. Finally, a specific example of wind forcing in a rectangular domain of constant depth is given. The model input is specified and the resulting output is shown.

As a final note, in the following chapters the information presented is sometimes repetitious, and this is partly a consequence of writing a technical document. However it does save the user the trouble of constantly refering to other parts of the document when trying to find some specific explanation.

2. SYMBOLS, CONSTANTS, AND COMMON BLOCKS

As the model has evolved and been applied to various problems, constants and variables sometimes have been added to or deleted from the program. However, this section describes the constants and variables of the COMMON block and MAIN program that are basic to all applications of the model.

The variables and arrays in the COMMON block are grouped into several labeled common subgroups:

COMMON/BLKCON/	Listing of constants.
COMMON/BLK1D/	Listing of one dimensional arrays.
COMMON/BLK2D/	Listing of two dimensional arrays.
COMMON/BLD3D/	Listing of three dimensional arrays.
COMMON/BDRY/	Listing of boundary value arrays.

In model application, the labeled commons are often put into separate computer files, and then referred to in the MAIN program and subroutines with INCLUDE statements. This facilitates any changes that might be made in the subsequent application. This chapter describes the constants and variables used in the COMMON block and MAIN program.

INDICES

I	Horizontal grid index in X direction, I=1,...,IM.
J	Horizontal grid index in Y direction, J=1,...,JM.
K	Vertical grid index for the sigma (nondimensional) coordinate system. K=1,...,KB. K=1 at the top where Z=0, and K=KB at the bottom where Z=−1.

The parameters specified in the common block are

IM	Maximum number of points in X direction.
JM	Maximum number of points in Y direction.

KB	Maximum number of points in Z direction.

The other parameters are straightforwardly derived from these and are used for dimensioning arrays.

CONSTANTS IN COMMON/BLKCON/

DTE	The external mode (barotropic) time step in seconds.
DTI	The internal mode (baroclinic) time step in seconds.
GRAV	The acceleration of gravity (=9.807 $m\ s^{-2}$).
IINT	The internal mode time step index (IINT=1,...,IEND). Some earlier versions of the model used INT.
IPRINT	The interval (in number of internal mode time steps) at which variables are printed.
TIME	The time in days and fractions of a day.
TPRNU	Weighting factor used in subroutine ADVT to calculate diffusive fluxes over grid spaces (=1.0). TURBULENT PRANDTL NUMBER
RAMP	The fraction of an inertial period (or sometimes the fraction of one day) that has elapsed since the start of the calculation. If RAMP $\geq$ 1.0, then RAMP = 1.0. It is used to spin up the forcing gradually over one inertial period to minimize spurious wave generation.
UMOL	Background diffusivity (=1.0E–4).

ONE DIMENSIONAL ARRAYS IN COMMON/BLK1D/

Z(KB)	The nondimensional depths in the sigma coordinate system are the levels at which the variables are evaluated. They vary from Z(1)=0 at the surface to Z(KB)=–1.0 at the bottom.

DZ(KB)	The nondimensional grid spacing in the vertical between the Z(K) levels. The last element is set equal to zero DZ(KB)=0.
ZZ(KB)	The sigma levels midway between the Z(K) levels. ZZ(K)=0.5*(Z(K)+Z(K+1))
DZZ(KB)	The grid spacing between the ZZ(K) levels. The last element is set equal to zero DZZ(KB)=0.
DZR(KB)	The inverse of the grid spacing. DZR(K)=1.0/DZ(K). The last element is set equal to zero DZR(KB)=0.

TWO DIMENSIONAL ARRAYS IN COMMON/BLK2D/

AAM2D(IM,JM)	The vertically integrated horizontal eddy viscosity term.
ALAT(IM,JM)	The latitude of a grid box centered on a depth H point.
ALON(IM,JM)	The longitude of a grid box centered on a depth H point.
ANG(IM,JM)	The angle between the two curvilinear axes for a grid box. It is $\pi/2$ for cartesian and spherical coordinates.
ART(IM,JM)	The area of a grid box centered at a depth point. ART(I,J)=DX(I,J)*DY(I,J)
ARU(IM,JM)	The area of a grid box centered at a U velocity point.
ARV(IM,JM)	The area of a grid box centered at a V velocity point.
CBC(IM,JM)	The coefficient of bottom friction. It may be calculated in MAIN and depends on depth and σ-level distribution but is at least set to a minimum value CBCMIN=0.0025.
COR(IM,JM)	The value of the coriolis parameter for each grid box centered on an H point. Earlier versions of the model use COR4(IM,JM), the value of the coriolis parameter divided by 4.

CURV2D(IM,JM)	Initially this array is set equal to COR. In subroutine ADVU this array is calculated to be the sum of COR and the vertical integral of the array CURV resulting from curvature terms in orthogonal curvilinear coordinates. Earlier versions of the model use CURV42D, the above value divided by 4.
D(IM,JM)	The instantaneous depth in meters of the water column in the external mode time step. D(I,J) = H(I,J)+EL(I,J)
DT(IM,JM)	The instantaneous depth of the water column in meters at each internal mode time step. DT(I,J) = H(I,J)+ET(I,J)
DUM(IM,JM)	The U velocity depth mask. It is set equal to one for ocean grid boxes and equal to zero for land grid boxes and those ocean grid boxes with land immediately to the left (lesser I). Then U(I,J)=U(I,J)*DUM(I,J).
DVM(IM,JM)	The V velocity depth mask. It is set equal to one for ocean grid boxes and equal to zero for land grid boxes and those ocean grid boxes with land immediately to the south (lesser J). Then V(I,J)=V(I,J)*DVM(I,J).
DX(IM,JM)	The grid spacing in the X direction for each grid box, in meters.
DY(IM,JM)	The grid spacing in the Y direction for each grid box, in meters.
EGB(IM,JM)	The value of EGF at the previous time step.
EGF(IM,JM)	The external mode sea level in meters averaged over the number of external mode time steps in the internal mode time step. It is calculated by increments using EGF(I,J)= EGF(I,J)+EL(I,J)*ISPI.
EL(IM,JM)	The surface elevation, in meters, as used in the external mode at the central time step.
ELB(IM,JM)	as above, but at previous (back) time step.
ELF(IM,JM)	as above, but at next (forward) time step.

ET(IM,JM)	The surface elevation in meters used in the internal mode at the central time step.
ETB(IM,JM)	as above, but at previous (back) time step.
ETF(IM,JM)	as above, but at next (forward) time step.
FLUXUA(IM,JM)	Array used to calculate terms in the horizontal advection arrays ADVUA and ADVVA in SUBROUTINE ADVAVE. Also used in calculating barotropic free surface by the continuity equation in MAIN.
FLUXVA(IM,JM)	as above.
FSM(IM,JM)	The sea level (and temperature, salinity, density, and vertical velocity) mask. It is set equal to zero for land grid squares and one for ocean grid squares. D(I,J)=D(I,J)*FSM(I,J)
H(IM,JM)	The bottom depth (ocean depth at rest) in meters for each grid box. Land values are typically set to a value between 0 and 1.
PSI(IM,JM)	The stream function, computed diagnostically.
TPS(IM,JM)	Temporary Storage Space array. A number of different variables are stored in this space at various places in the program.
UA(IM,JM)	The vertically averaged (barotropic) velocity in the X-direction used in the external mode calculations, at the central time step, in ms^{-1}.
UAB(IM,JM)	as above, but at the previous (back) time step.
UAF(IM,JM)	as above, but at the next (forward) time step.
VA(IM,JM)	The vertically averaged (barotropic) velocity in the Y-direction used in the external mode calculations, at the central time step, in ms^{-1}.
VAB(IM,JM)	as above, but at the previous (back) time step.
VAF(IM,JM)	as above, but at the next (forward) time step.

WSSURF(IM,JM) The salinity flux at the surface.

WTSURF(IM,JM) The heat flux at the surface.

WUBOT(IM,JM) The X-direction momentum flux at the bottom. It is calculated in subroutine ADVAVE for the external mode, and in subroutine PROFU for the internal mode, with a quadratic resistance law.

WVBOT(IM,JM) The Y-direction momentum flux at the bottom. It is calculated in subroutine ADVAVE for the external mode, and in subroutine PROFV for the internal mode, with a quadratic resistance law.

WUSURF(IM,JM) The X-direction momentum flux at the surface. It is the negative of the surface wind stress divided by the water density, and so it is negative for westerly winds. The wind stress is in Nt m^{-2} and the constant water density 1024 Kg m^{-3} is used, so that WUSURF has dimensions $m^2 s^{-2}$ and is of typical magnitude 10^{-4}.

WVSURF(IM,JM) The Y-direction momentum flux at the surface. It is the negative of the surface wind stress divided by the water density, as above.

THREE DIMENSIONAL ARRAYS IN COMMON/BLK3D/

A(IM,JM,KB) An array used as a holding space by use of an EQUIVALENCE statement in various subroutines. This array is also used directly for calculations in the various vertical profile subroutines, PROFU, etc.

AAM(IM,JM,KB) Horizontal kinematic viscosity. The array is initialized with typical values of 50 – 2000 $m^2 s^{-1}$.

C(IM,JM,KB) An array used as a holding space by use of an EQUIVALENCE statement in various subroutines. This array is also used directly for calculations in the various vertical profile subroutines, PROFU, etc.

DTEF(IM,JM,KB)	An array used in subroutine PROFQ to calculate vertical profiles.
KH(IM,JM,KB)	Vertical diffusivity mixing coefficient for temperature and salinity (real variable).
KM(IM,JM,KB)	Vertical kinematic viscosity momentum mixing coefficient (real variable).
KQ(IM,JM,KB)	Vertical mixing coefficient for turbulence (real variable).
L(IM,JM,KB)	Turbulent length scale (real variable).
Q2(IM,JM,KB)	Twice the turbulent kinetic energy.
Q2B(IM,JM,KB)	as above, but at previous (back) time step.
Q2L(IM,JM,KB)	The product of twice the turbulent kinetic energy times the turbulence length scale, Q2L(I,J,K)=Q2(I,J,K)*L(I,J,K).
Q2LB(IM,JM,KB)	as above, but at previous (back) time step.
RHO(IM,JM,KB)	Density divided by 1000 with 1.025 subtracted. This is done to gain precision since only the gradient of density enters into the calculations. Density calculations are made in subroutine DENS. To obtain the value in $Kg\ m^{-3}$ from the value computed in DENS, add 1.025 and then multiply the result by 1000.
RHOF(IM,JM,KB)	Density at forward time step, defined as above, used in some earlier versions of the model.
RMEAN(IM,JM,KB)	Area averaged density, defined as above.
S(IM,JM,KB)	Salinity in parts per thousand (ppt) with 35 ppt subtracted from real value.
SB(IM,JM,KB)	as above, but at previous (back) time step.
SMEAN(IM,JM,KB)	The average salinity, as above.
T(IM,JM,KB)	Celsius Temperature, with 10 C^o subtracted.

TB(IM,JM,KB)	as above, but at previous (back) time step.
TMEAN(IM,JM,KB)	The average Celsius temperature, as above.
U(IM,JM,KB)	The horizontal velocity in the X-direction, in ms^{-1}.
UB(IM,JM,KB)	as above, but at previous (back) time step.
UF(IM,JM,KB)	as above, but at next (forward) time step.
V(IM,JM,KB)	The horizontal velocity in the Y-direction, in ms^{-1}.
VB(IM,JM,KB)	as above, but at previous (back) time step.
VF(IM,JM,KB)	as above, but at next (forward) time step.
VH(IM,JM,KB)	An array used in subroutines PROFQ, PROFU, and PROFV to calculate vertical profiles.
VHP(IM,JM,KB)	as above.
W(IM,JM,KB)	Vertical velocity in σ-coordinate system.

BOUNDARY VALUE ARRAYS IN COMMON/BDRY/

COVRHE(JM)	Array used in calculating free gravity wave radiation condition on eastern boundary.
COVRHN(IM)	Array used in calculating free gravity wave radiation condition on northern boundary.
ELE(JM)	Sea level in meters on eastern boundary, EL(IM,J).
ELN(IM)	Sea level in meters on northern boundary, EL(I,JM).
ELS(IM)	Sea level in meters on southern boundary, EL(I,1).
SBE(JM,KB)	Salinity on eastern boundary, SB(IM,J,K), with 35 ppt subtracted from real values.

SBN(IM,KB)	Salinity on northern boundary, SB(I,JM,K), as above.
SBS(IM,KB)	Salinity on southern boundary, SB(I,1,K), as above.
TBE(JM,KB)	Temperature on eastern boundary, TB(IM,J,K), with 10 C^o subtracted from real values.
TBN(IM,KB)	Temperature on northern boundary, TB(I,JM,K), as above.
TBS(IM,KB)	Temperature on southern boundary, TB(I,1,K), as above.
UABE(JM)	Barotropic velocity on eastern boundary, UAB(IM,J), in ms^{-1}.
VABN(IM)	Barotropic velocity on northern boundary, VAB(I,JM), as above.
VABS(IM)	Barotropic velocity on southern boundary, VAB(I,1), as above.

OTHER VARIABLES NOT IN COMMON BLOCK

CONSTANTS

AAA	A constant value (typically 500 – 2000 $m^2\ s^{-1}$) sometimes used to initialize the horizontal kinematic viscosity array, AAM(I,J,K)=AAA.
ALPHA	Constant used in weighted averaging the sea level or pressure gradient forcing over the three time steps, in MAIN (typically ALPHA= 0.225).
CBCMIN	Minimum value of the bottom friction term (usually =0.0025).
DAYI	The fraction of one day occupied by one second, or inverse day (=1/86400).

DTE2	Twice the external mode time step (=DTE*2).
DTI2	Twice the internal mode time step (=DTI*2).
FACTOR	The number of internal mode time steps in one day. FACTOR=24*3600/IFIX(DTI)
GSMOD	A function used in some applications to apply a time modulation to an inflow boundary condition in subroutine BCOND. (Gulf Stream modulation)
HORCON	A nondimensional constant used in calculating the horizontal eddy diffusivity (typically 0.01 – 0.10).
IDAYS	The length of time in days that the model simulation will run.
IEND	The total number of internal mode time steps of the model run, IEND=IDAYS*24*3600/IFIX(DTI).
IPRINT	The interval (in number of internal mode time steps) at which variables are printed. It is set by IPRTD1 or IPRTD2, below.
IPRTD1	Time interval in days at which data is written to the standard out. Initially, the model sets IPRINT=IPRTD1*24*3600/IFIX(DTI). Some model versions use the variable IPRNT1.
IPRTD2	Time interval in days at which data is written to the standard out, later in the model run, see ISWTCH, below. Some model versions use IPRNT2.
ISPADV	The frequency interval (in external mode time steps) at which the advection subroutine ADVAVE is called.
ISPI	The ratio of the external to internal time step ISPI=1.0/FLOAT(ISPLIT). It is a real number.
ISP2I	Half the value of ISPI. It is a real number.
ISPLIT	The ratio of the internal to external time steps (=DTI/DTE).

ISWTCH	The number of internal mode time steps after which the model will switch to printing out data at a different interval. It is used in the print section at the end of the MAIN program, in the form IF(IINT.GT.ISWTCH) IPRINT=IPRTD2*24*3600/IFIX(DTI).
MODE	The type of calculation the model performs: MODE=2, performs a two dimensional calculation (bottom stress calculated in ADVAVE). MODE=3, performs a three dimensional calculation (bottom stress calculated in PROFU, PROFV). MODE=4, performs a diagnostic calculation, where the temperature and salinity at each point are held constant in time.
NREAD	The number of data sets read over when reading input from a restart file.
PERIOD	The period of time over which the forcing is ramped or spun up, expressed in days or fractions of a day. It is often the inertial period but is sometimes set equal to one day, or longer as required.
RE	The radius of the earth ($= 6.371 \times 10^6$ m).
SMALL	A small number (1.E–10) added to the denominator of some fractions to insure division by zero does not occur. Used in determining L from Q2LB and Q2B in the initialization.
SMOTH	Constant used in the time filter for the Leapfrog time scheme (typically 0.05 – 0.10).
TIME0	The value of TIME in days at the start of the model calculation (usually TIME0=0.). If the model run is a restart after some time has elapsed, TIME0 will be this elapsed time in days.
VAMAX	The maximum value of VA(I,J) for the domain at the given time step. It is used as a stability check to stop the program if the values become too high.
VAMIN	The minimum value of VA(I,J), as above.

TWO DIMENSIONAL ARRAYS

ADVUA(IM,JM)	Horizontal advection and viscosity in the barotropic equation for UA, calculated in subroutine ADVAVE.
ADVVA(IM,JM)	as above, but for the VA equation.
ADVUU(IM,JM)	Vertical integral of the horizontal dispersion terms used in barotropic UA momentum equation, calculated in SUBROUTINE ADVU.
ADVVV(IM,JM)	As above, but fot VA equation, and calculated in SUBROUTINE ADVV.
SSURF(IM,JM)	The surface salinity, used in subroutine ADVT in the calculation of salinity advection. At initialization it is set to SSURF(I,J)=SB(I,J,1).
TRNU(IM,JM)	X-component of the vertical integral of baroclinic pressure gradient (vertical integral of DRHOX), calculated in SUBROUTINE BAROPG.
TRNV(IM,JM)	Y-component, as above.
TSURF(IM,JM)	The surface temperature, used in subroutine ADVT in the calculation of temperature advection. At initialization it is set equal to TSURF(I,J)=TB(I,J,1).
TX(IM,JM)	The X-direction wind stress used in some model applications to read a wind stress in, say dynes cm^{-2} from which WUSURF is obtained.
TY(IM,JM)	The Y-direction wind stress, as above.
UT(IM,JM)	The average value of the barotropic velocity UA multiplied by the depth D, averaged over the internal mode time step.
UTF(IM,JM)	As above, at the forward time step.
VT(IM,JM)	The average value of the barotropic velocity VA multiplied by the depth D, averaged over the internal mode time step.

VTF(IM,JM)	As above, at the forward time step.

THREE DIMENSIONAL ARRAYS

CURV(IM,JM,KB)	Array that contains the effects of the curvature terms in curvilinear coordinates and the Coriolis effect. It is used in the subroutines ADVU and ADVV. Earlier versions of the model used the array CURV4, the above value divided by 4.
DRHOX(IM,JM,KB)	X-component of the internal baroclinic pressure gradient, due to density differences, calculated in SUBROUTINE BAROPG.
DRHOY(IM,JM,KB)	as above, Y-component.
PROD(IM,JM,KB)	Turbulent kinetic energy production rate.

The model is very general in both the types of problems for which it can obtain solutions, and the way in which the computer code is applicable to each problem. Generally, it will only be necessary to change the MAIN program and subroutine BCOND to configure the model to a specific problem. Here we shall describe the necessary input the user must supply to the model (a specific example is provided in the chaper describing the box model test case). The model uses the MKS (Meter – Kilogram – Second) system of units. Once some problem dependent values (such as depth H(I,J)) are supplied, the model uses this input to derive other necessary parameters (such as the land mask FSM(I,J)). Other parameters such as HORCON, a constant used in calculating the horizontal eddy diffusivity, may be changed by the user for specific experiments, but are not otherwise considered as user supplied input.

Most variables are initialized to zero at the very start of the MAIN program. This is necessary even if they will be assigned a different value later, because some calculations may exceed the bounds of an array, and take an adjacent value in the COMMON block, before being set to zero in subsequent calculations. If the model is to be spun up from rest, the initial velocities and sea level will be set to zero. If the model is to be restarted after some time, or if an input file has been used to initialize the model, the necessary values are read in with the READ statements at the start of the MAIN program. It would be convenient for the user to append the subroutines DEPTH and DENS to any initialization program. Their use in the model initialization is discussed below.

When initialization files are used to start the model, they usually have been read in using the following conventions.

```
      READ(40) Z,ZZ,DZ,DZZ,ALON,ALAT,DX,DY,H,COR4,ANG,
     1 ART,ARU,ARV,TMEAN,SMEAN,RMEAN,TBN,TBE,TBS,SBN,SBE,SBS,
     2 ELN,ELE,ELS,VABN,UABE,VABS

      READ(50) TIME0,UB,VB,UAB,VAB,TB,SB,ELB,Q2B,Q2LB,KM,KH

      READ(60) WUSUFR,WVSURF,WTSURF
```

The unit 40 contains the physical parameters and climatological data form the problem. The temperature and salinity data is usually taken from the Levitus climatology. Typically, the unit 50 initialization file includes the velocities, temperatures, and salinities created by starting the model from rest and spinning it up over a long enough time to stabilize the variables. The unit 60 contains the surface momentum fluxes (often derived from the Hellerman wind stress climatology) and surface heat flux. The user can modify the initialization files to suit the problem. Since the model uses the centered in time (leapfrog) time step, two levels are necessary to initialize the model. If the model is started from rest this is taken

care of for the velocities when all variables are set to zero at the start of the program. The MAIN program includes the code to set the variables of the central time step equal to those of the back time step (for example, T(I,J,K)=TB(I,J,K)). If both time levels were saved to a restart file, then both can be read in. Restart files are discussed at the end of this chapter.

When the model is started from rest, it is sufficient to specify the variables discussed below. The other variables such as the velocities will have been initialized to zero at the start of the MAIN program.

1. The Problem Dimensions

It is first necessary to specify the size of the domain. These dimensions are

IM	Maximum number of points in the X direction, I=1,...,IM
JM	Maximum number of points in the Y direction, J=1,...,JM
KB	Maximum number of points in the vertical, K=1,...,KB.

These are specified in the PARAMETER statements in the COMMON block. Other parameters such as KBM1=KB–1 are straightforwardly derived and are used for dimensioning arrays. Subroutine DEPTH establishes the vertical resolution with a log distribution at the top and bottom, and a linear distribution in between. This subroutine is called only once when setting up the model domain at the start of the MAIN program. In some applications of the model, the vertical resolution is determined by subroutine DEPTH as part of a prior initialization, and then read as input to MAIN. When subroutine DEPTH is used to determine the vertical spacing, there should be at least six (KB = 6) vertical points for the routine to work properly, and this would be an absolute minimum. The subroutine DEPTH tends to put more points near the surface to resolve the mixed layer. It is the users responsibility to decide how many points are necessary to do this for any specific application. Model applications have been run for the Gulf of Mexico with KB = 12 and 18, and for the North Atlantic Gulf Stream region with KB = 12 and 15. In a call to DEPTH, the subroutine returns the vertical spacing of points in the σ coordinate system. These σ levels vary from Z(1)=0 at the surface to Z(KB)=–1.0 at the bottom. The subroutine also returns the distances between each σ level, DZ(K); the distribution of the midpoints between each level, ZZ(K); and the distance between midpoints of each level, DZZ(K). The last values of the spacing arrays are set equal to zero, DZ(KB)=DZZ(KB)=0. If the user desires to specify the particular vertical spacing, then the arrays Z,DZ,ZZ, and DZZ must be explicitly given.

2. It is necessary to specify the grid spacing in the X and Y directions. This is the grid spacing for each grid box centered on a depth point for the Arakawa-C grid, and the distances are in meters. These are specified as two arrays.

DX(IM,JM)	The grid spacing in the X direction for each grid box, in meters.
DY(IM,JM)	The grid spacing in the Y direction for each grid box, in meters.

The model or an initialization program then uses these values to calculate the areas of the grid boxes.

ANG(IM,JM)	The angle between the two curvilinear axes for a grid box. It is $\pi/2$ for cartesian and spherical coordinates.
ART(IM,JM)	The area of a grid box centered on a depth point
ARU(IM,JM)	The area of a grid box centered on a U velocity point.
ARV(IM,JM)	The area of a grid box centered on a V velocity point.

The areas of the grid squares are made by the calculations

```
ART(I,J)=DX(I,J)*DY(I,J)
ARU(I,J)=0.25*(DX(I,J)+DX(I-1,J))*(DY(I,J)+DY(I-1,J))
ARV(I,J)=0.25*(DX(I,J)+DX(I,J-1))*(DY(I,J)+DY(I,J-1))
```

with the special cases of

```
ARU(1,J)=ARU(2,J)
ARV(1,J)=ARV(2,J)
ARU(I,1)=ARU(I,2)
ARV(I,1)=ARV(I,2).
```

The most common coordinate systems used are cartesian and spherical, but other orthogonal curvilinear coordinate systems can be set up for particular domains using a grid generation program.

3. The value of the coriolis parameter must be specified for each grid box. This is the array COR(IM,JM). This feature allows the model to be used for large ocean domains, while for

smaller domains this value can be the same for each grid box. Some earlier versions of the model use the value of the coriolis parameter divided by 4, COR4(IM,JM), and have the dynamical equations programmed accordingly.

Some versions of the model have the arrays ALAT(IM,JM) and ALON(IM,JM), which give the latitude and longitude of each grid box. These arrays are useful for recalling the location of each grid box for graphics work, and calculating the coriolis parameter, but are not used directly in the model calculations.

4. The bottom topography is specified with the ocean depth in H(IM,JM). For some applications it may be advisable to filter the bottom topography with a Shapiro (1970) filter to remove $2\Delta x$ noise. This depth array is specified over the entire domain, both land and ocean. For the land grid squares, H is set to a value of between 0.0 and 1.0, typically H(I,J)=0.5. The finite difference calculations proceed over all the grid squares of the domain, both land and ocean, and the values calculated at land points are subsequently set to zero with appropriate land and velocity masks. These masks for land and velocity then make use of the criterion that $0.0 \leq H(I,J) \leq 1.0$ to define a land grid square. However, the time step calculations make use of division by H(I,J), and so everywhere H must have a nonzero value. When the model is run in the baroclinic mode with, say, 10 vertical levels (KB=10), then the minimum depth should be several meters, to avoid computational instabilities. This may also depend on the range of depths over the model domain. When the model was used to simulate hurricane landfalls in the Gulf of Mexico with KB = 18, the minimum depth used was 10 m. For barotropic (vertically integrated) mode calculations, the shallowest depth could be about two meters. However, these features are very problem specific, and depend a great deal on the model domain and the forcing.

5. The Temperature, Salinity, and Density

It is necessary to specify the initial and mean temperatures and salinities as input to the model.

TMEAN(IM,JM,KB)	The area averaged temperature in degrees Celsius, with 10 C^o subtracted before input to the model. It is used in the advection subroutine ADVT.
SMEAN(IM,JM,KB)	The area averaged salinity in parts per thousand (ppt), with 35 ppt subtracted before input to the model. It is used in the advection subroutine ADVT.
TB(IM,JM,KB)	The initial temperature in degrees Celsius, with 10 C^o subtracted before input to the model.

SB(IM,JM,KB)	The initial salinity in parts per thousand, with 35 ppt subtracted before input to the model.

The modeler should subtract 10 C^{o} from the temperature and 35 ppt from the salinity as part of the model initialization. This results in additional precision, since for most calculations only the gradients of temperature and salinity are taken. However, in subroutine DENS, these values are added back to the temperature and salinity before the density is calculated. All calculations on the CRAY-YMP are performed using 64 bit precision. When the model is run on a machine with 32 bit precision, the calculations in subroutine DENS should be made in double precision.

The temperature and salinity may be taken from the Levitus climatology or specified appropriately for the problem. Generally, when the model is initialized from a climatology (as opposed to restarting the model after some period of time), the arrays TMEAN and TB will be set to the same initial values, and the arrays SMEAN and SB will be set to the same initial values.

RMEAN(I,J,K)	The area averaged density divided by 1000 with 1.025 subtracted. To obtain the value in Kg m^{-3} form the value computed in DENS, add 1.025 and then multiply the result by 1000.

The area averaged density must be supplied to initialize the model. It is initially determined by a call to subroutine DENS using the initial temperature and salinity as input parameters. When an initialization program is used to supply input to the model, this program would have to include the subroutine DENS. The density RHO(I,J,K) is the returned parameter from DENS and then the mean density is then defined by setting RMEAN(I,J,K)=RHO(I,J,K) . The mean density is subtracted from the density at the beginning of subroutine BAROPG, which computes the baroclinic pressure gradient. This results in some increase in accuracy of the calculations. At the end of subroutine BAROPG the mean density is added back to the value of the density. Here we note that subroutine DENS computes the density divided by 1000 with 1.025 subtracted. This is done to gain precision since only the gradient of density enters into the calculations. To obtain the density value in Kg m^{-3} from the value computed in subroutine DENS, add 1.025 and then multiply the result by 1000.

6. The boundary value arrays should be specified. The boundaries and boundary value arrays are dependent on the particular problem. The model version supplied assumes that the western boundary is a land boundary, while the north, south, and east boundaries can be open and have boundary value arrays dimensioned for them. The user can configure the model to different boundary geometries. If one of the north, south, or east boundaries is a

land boundary, the boundary value arrays need not be used. At a land boundary the masking is all that is necessary to determine the correct boundary values there. If the model requires an open western boundary, the user will have to define appropriate boundary value arrays. The sea level, temperature, salinity, and velocity boundary value arrays for the northern, eastern, and southern boundaries are

ELE(JM) Sea level EL(IM,J) on eastern boundary.

ELN(IM) Sea level EL(I,JM) on northern boundary.

ELS(IM) Sea level EL(I,1) on southern boundary.

TBN(IM,KB) Temperature in C° on northern boundary, with 10 C° subtracted from real values.

TBS(IM,KB) Temperature in C° on southern boundary, as above.

TBE(JM,KB) Temperature in C° on eastern boundary, as above.

SBN(IM,KB) Salinity in ppt on northern boundary, with 35 ppt subtracted from the real values.

SBS(IM,KB) Salinity in ppt on southern boundary, as above.

SBE(JM,KB) Salinity in ppt on eastern boundary, as above.

UABE(JM) Barotropic velocity UAB(IM,J) on eastern boundary.

VABN(IM) Barotropic velocity VAB(I,JM) on northern boundary.

VABS(IM) Barotropic velocity VAB(I,1) on southern boundary.

For purposes of initialization, it is probably best to set these boundary value arrays to the appropriate values of climatology, the same values for TB(I,J,K) with 10 C° subtracted, and SB(I,J,K) with 35 ppt subtracted. The boundary conditions vary greatly with the problem, and this will be discussed further in the section on subroutine BCOND.

7. The wind stress

The surface momentum flux must be specified. These are the arrays

WUSURF(IM,JM)	The X–direction momentum flux at the surface. It is the negative of the surface wind stress divided by the water density, and so it is negative for westerly winds. The wind stress is in Nt m^{-2} and the constant water density 1024 Kg m^{-3} is used, so that WUSURF has dimensions $m^2 s^{-2}$ and is of typical magnitude 10^{-4}.
WVSURF(IM,JM)	The Y–direction momentum flux at the surface. It is the negative of the surface wind stress divided by the water density, and so it is negative for northerly winds, as above.

In some applications of the model it is convenient to read in the data in the centimeter – gram – second units of dynes cm^{-2}. When this value is divided by the water density 1.0 gm cm^{-3} the result is of magnitude one $cm^2 s^{-2}$. This data is usually read into the model with the arrays.

TX(IM,JM)	The wind stress in the X direction divided by the water density, in units of $cm^2 s^{-2}$.
TY(IM,JM)	The wind stress in the Y direction divided by the water density, in units of $cm^{-2} s^{-2}$.

and the values of WUSURF and WVSURF can be obtained from them by changing the sign and multiplying by the scaling factor 10^{-4}. These arrays TX and TY do not appear in all versions of the model, but can be added by the user.

8. The surface temperature and salinity fluxes

These surface fluxes are presently set to zero in the time integration in the MAIN program. The user can comment this out and supply the input if desired.

WTSURF(IM,JM)	The surface temperature flux.

WSSURF(IM,JM)	The surface salinity flux.

9. The Time Steps

DTE	The external mode time step in seconds.
DTI	The internal mode time step in seconds.
ISPLIT	The ratio of the internal to external mode time steps (=DTI/DTE), expressed as an integer.

The internal (baroclinic) time step DTI and external (barotropic) time step DTE must be set appropriate to the problem. The external time step is the shorter and its value is governed by the CFL computational stability criterion, where this time step must be shorter than the time it takes the free surface gravity wave to travel between any two grid points. The model code includes the calculation of the minimum time step to meet the CFL criterion for each grid square, based on the grid spacing and ocean depth. The resulting data is stored in the array TPS (Temporary Storage Space). In practice, after considering the external mode CFL criterion, set values for DTI and ISPLIT. For example, to obtain an external mode time step of 30 seconds, and an internal mode time step of 600 seconds, set DTI=600.0 and ISPLIT=20, so that the model will subsequently calculate the value DTE=30.0 with the command DTE=DTI/FLOAT(ISPLIT).

10. The TIME and print intervals

The model simulation time is measured in days and fractions of a day, and two parameters must be set to the correct value.

TIME	The elapsed model time in days and fractions of a day.
TIME0	The value of TIME in days at the start of the model calculation. If the model run is started from a restart file, TIME0 will be the time read in from the restart file.

Initially the model is set with TIME=0.0. When information is written to a file at the end of some time interval, the variable TIME is usually the first variable written to the file. If the model is restarted after some time, be sure that the model uses (or reads in from an input file) the appropriate value TIME0. The model then sets TIME=TIME0.

The model is integrated forward in time counting the internal mode time steps, using the loop DO 9000 IINT=1,IEND, and so it is necessary to specify

IEND	Total number of internal mode time steps.

For example, if the model is to be run for ten days, with the internal mode time step DTI=600.0 seconds, then set IEND=1440. In the current version of the model, the user specifies

IDAYS	Time in days of model simulation.

and the model calculates the value of IEND with the statement

IEND = IDAYS*24*3600/IFIX(DTI).

The intervals at which files are to be printed out or saved must be specified. These specifications are based on the number of internal mode time steps needed to cover the time interval. The model comes with two print intervals at which data can be written to the standard out (unit 6)

IPRTD1	Time interval in days at which data is written to the standard out, used at start of model. The model sets IPRINT=IPRTD1*24*3600/IFIX(DTI).
IPRTD2	Time interval in days at which data is written to the standard out, later in the model run.
ISWTCH	The number of internal mode time steps after which the model will switch to printing out data at a different interval. It is used in the printing section at the end of the main program, in the form IF(IINT.GT.ISWTCH) IPRINT=IPRTD2*24*3600/IFIX(DTI).

It is useful to have two print intervals for short debugging runs, but otherwise it is only necessary to use one (set both IPRTD1 and IPRTD2 to the same number of days).

In the version of the model at INO, the intervals at which files are be written can be specified with the variables

ISVEL	Interval at which data is saved to a file.
ISVTS	Interval at which data is saved to a file.

In the version of the model at INO, these values are specified in days and the model then calculates the equivalent number of internal mode time steps, with the command

ISVEL=ISVEL*24*3600/IFIX(DTI).

The exact specifics of how the print and file save intervals are set may depend on the exact version of the model. The user can easily adapt the proceedure to save as many files as necessary at whatever intervals are desired.

11. The MODE or type of calculation to be performed must be specified. It can be one of the three values

MODE=2.	Model performs a two dimensional (X,Y) vertically integrated calculation for the average velocities UA(I,J), VA(I,J), and the sea level EL(I,J).
MODE=3	Model performs a fully three dimensional calculation.
MODE=4	Model performs a three dimensional diagnostic calculation. The temperature and salinity (and hence density) at each grid point are constant, so there is no advection of temperature, salinity, or density.

12. The vertical kinematic viscosity KM, vertical heat diffusivity KH, twice the turbulent kinetic energy Q2B, turbulence length scale L, and Q2LB(I,J,K)=Q2B(I,J,K)*L(I,K,J) can be initialized with zero values or set to some small value on the order 10^{-4}. The horizontal kinematic viscosity AAM(I,J,K) can be set to some value (typically 50 – 2000 $m^2 s^{-1}$) depending on the dynamics of the problem and the grid spacing.

13. The Spin up time or PERIOD over which the model forcing is increased linearly from zero to its actual value should be specified.

PERIOD	The period of time in days over which the forcing is spun up. It is usually set to the inertial period in days, or sometimes one day, but can be set to longer

	times to spin up special problems (such as a model with very shallow depths).
RAMP	The factor that increases linearly from zero to one to spin up the forcing, RAMP=TIME/PERIOD.

The wind stress forcing and baroclinic pressure gradient forcing in the numerical integration of the MAIN program, and the inflow boundary velocities in subroutine BCOND are multiplied by the factor RAMP.

14. For long time simulations of the model, it may be desirable to write model data to a restart file. It is best to write binary data files for greater precision. Since the model uses a centered difference (leapfrog) time step, two time levels are necessary for a 'seemless' restart. This can be done with the command.

```
      WRITE(77) TIME,
     1 CURV42D,WUBOT,WVBOT,AAM2D,UA,UAB,VA,VAB,EL,ELB,
     2 ET,ETB,EGB,UTB,VTB,U,UB,W,V,VB,T,TB,S,SB,RHO,ADVUU,
     3 ADVVV,ADVUA,ADVVA,KM,KH,KQ,L,Q2,Q2B,AAM,Q2L,Q2LB
```

4. The MAIN Program

The MAIN program will vary somewhat depending on which version of the model is being used, and its application to a specific problem, but it consists of two logical parts. The first is the initialization, where the model parameters are prepared for the specific problem. The second logical part is the numerical integration of the model forward in time, which is done by the loop DO 9000 IINT=1,IEND. We shall discuss these parts separately.

The model form will also depend on whether it was optimized for a supercomputer with an array processor. In the latter case two arrays can be set equal with a single statement and it need not be enclosed in a DO loop. Then the statement numbers may vary somewhat with the specific version of the model. The statement numbers in the model initialization will vary with the application, depending on exactly what must be specified. However, the statement numbers required for the processing of the model numerical integration of the MAIN program are standard features and do not differ significantly between versions.

Part 1. The Initialization

This is the part of the model that will vary most for any specific application. The necessary initialization data can be included here, or written to a separate initialization data file and then read into the MAIN program for each model run. This latter method would be the most efficient if the model is to be run many times.

First the COMMON block is given explicitly or called from a separate file with an INCLUDE statement. The MAIN program then declares and dimensions several more arrays and constants. Most of these are physical parameters. A complete listing and description of the arrays and constants in the COMMON block and the other arrays and constants used in the MAIN program is given in Chapter 2, Symbols, Constants, and Common Blocks.

At the start of the MAIN program, most of the arrays are initialized to zero, even though other values will be assigned later in the program. This is necessary because the code was written for efficiency on vector processing computers, where calculations over the domain sometimes exceed the bounds of the array. Values are then taken from adjacent arrays stored in the COMMON block, and so all variables must be initialized to some value at the very beginning to avoid an 'undefined variable' error message. This initialization is done with Fortran DO loops or DATA statements. It is recommended that DO loops be used since initializing large arrays with DATA statements is very inefficient on some machines.

Next, the individual problem specifications are supplied appropriate to the problem. The model reads in the required data to start the model from initialization or restart files, or specifies it in this section of the MAIN program. These are described in detail in the section on User Supplied Input, and some discussion is repeated here. Basically, the time steps and print and file save intervals are specified, and any other model parameters are specified for the physical application.

If the model is to be spun up from rest, the initial velocities and sea level will be set to zero. When initialization files are used to start the model, they usually are read in using the following conventions

```
      READ(40) Z,ZZ,DZ,DZZ,ALON,ALAT,DX,DY,H,COR4,ANG,
     1 ART,ARU,ARV,TMEAN,SMEAN,RMEAN,TBN,TBE,TBS,SBN,SBE,SBS,
     2 ELN,ELE,ELS,VABN,UABE,VABS

      READ(50) TIME0,UB,VB,UAB,VAB,TB,SB,ELB,Q2B,Q2LB,KM,KH

      READ(60) WUSUFR,WVSURF,WTSURF.
```

The unit 40 contains the physical parameters and climatological data form the problem. The temperature and salinity data is usually taken from the Levitus climatology. Typically, the unit 50 initialization file includes the velocities, temperatures, and salinities created by starting the model from rest and spinning it up over a long enough time to stabilize the variables. The unit 60 contains the surface momentum fluxes (often derived from the Hellerman wind stress climatology) and surface heat flux. The user can modify the initialization files to suit the problem. Since the model uses the centered in time (leapfrog) time step, two levels are necessary to initialize the model. If the model is started from rest this is taken care of for the velocities when all variables are set to zero at the start of the program. The MAIN program includes the code to set the variables of the central time step equal to those of the back time step (for example, T(I,J,K)=TB(I,J,K)), and this is discussed below.

For long time simulations of the model, it may be desirable to write model data to a restart file. It is best to write binary data files for greater precision. Since the model uses a centered difference (leapfrog) time step, two time levels are necessary for a 'seemless' restart. The restart files are read over and the values from the last (NREAD–th) one is used to initialize the model with the command.

```
      DO 10 N=1,NREAD
      READ(77) TIME0,
     1 CURV42D,WUBOT,WVBOT,AAM2D,UA,UAB,VA,VAB,EL,ELB,
     2 ET,ETB,EGB,UTB,VTB,U,UB,W,V,VB,T,TB,S,SB,RHO,ADVUU,
     3 ADVVV,ADVUA,ADVVA,KM,KH,KQ,L,Q2,Q2B,AAM,Q2L,Q2LB
     10 CONTINUE
```

The following describes the most common form of the initialization part of the MAIN program. The processes are described with as little reference to Fortran statement numbers as possible.

The eddy diffusivity is initialized to a constant value, typically AAA = 500 – 2000 $m^2 s^{-1}$ with the statement AAM(I,J,K)=AAA.

Any data that is not already in the MKS system should be converted to this system. For example, if the wind stress is in dynes cm^{-2}, it can be multiplied by the factor 1.E–4 to convert it to $m^2 s^{-2}$ which is wind stress divided by water density, TX(I,J)=TX(I,J)*1.E–4.

The surface temperature and salinity arrays (used in heat and salinity flux calculations) are initialized with the statements

```
TSURF(I,J)=TB(I,J,1)

SSURF(I,J)=SB(I,J,1).
```

The surface elevation used in the internal mode calculations is initialized with the statement DT(I,J)=H(I,J)+ELB(I,J).

The command TIME=TIME0 sets the TIME variable to the value at the last restart file time.

The inverse of the vertical grid spacing is calculated with the command DZR(K)=1./DZ(K) for use in subsequent calculations. This is now done in subroutine DEPTH, but some earlier versions of DEPTH did not make this calculation.

The variable PERIOD is that time in days or fractions of a day over which the model is spun up with the RAMP variable. It is often calculated as the inertial period for the latitude of the center grid square of the domain but can be a longer period if required by the problem.

The Coriolis parameter is calculated for each grid square of the domain and stored in the array COR(I,J). Previous versions of the model used the value of the Coriolis parameter divided by 4, stored in COR4(I,J). The statement CURV2D(I,J)=COR(I,J) sets the array CURV2D equal to the value of the Coriolis parameter. In the subroutine ADVU the array CURV2D is calculated to be the sum of COR and the vertical integral of a curvature array CURV, resulting from curvilinear coordinate systems.

The land and velocity masks are determined from the bottom topography in three DO loops. We recall that the model grid extends over all land and ocean points. Land grid points are distinguished by having a depth between zero and one ($0 < H(I,J) < 1.0$). They must be assigned some nonzero value because the calculations that include division by the depth take place over the entire grid of land and ocean points. The loop DO 30 J=1,JM determines the land mask FSM(I,J). Using the above depth criterion, this array is assigned the value 0.0 for land grid squares and 1.0 for ocean grid squares. The grid squares for the model are those for the Arakawa–C grid. The depth (and temperature, salinity, vertical velocity, and sea level) points are at the center of the grid box, the U velocity points are the midpoint of the left side, and the V velocity points are at the bottom (south). The 'masking' or setting to zero of some value over land points is usually done in suboutine BCOND by array multiplication of the variable times the mask FSM, eg., EL(I,J)=EL(I,J)*FSM(I,J).

The loop DO 35 J=1,JM determines the V velocity mask DVM(I,J). If the grid square is a land grid or an ocean grid with a land grid immediately to the south (lesser J value), the value is zero. Otherwise, the mask is set equal to 1.0. The velocities are usually masked in subroutine BCOND by array multiplication, eg., VF(I,J,K)=VF(I,J,K)*DVM(I,J).

The U velocity mask DUM(I,J) is determined in the loop DO 40 J=1,JM If the grid square is a land grid or an ocean grid with a land grid immediately to the left, the value is

zero. Otherwise the mask is set equal to 1.0. The velocities are usually masked in subroutine BCOND by array multiplication, eg., UA(I,J)=UA(I,J)*DUM(I,J).

Some versions of the model have the statements DUM(1,J)=0.0 and DVM(1,J)=0.0. They make the entire western boundary a land boundary which was used in some Gulf Stream models to put a wall across the Florida Straits. It is a problem dependent feature.

The call to subroutine VABFIX is used in some models to specify temperatures, salinities, and inflow velocities across an open boundary. This is done with models of the Gulf Stream region to specify an inflow to the domain along the northern boundary across the continental shelf, in order to obtain Gulf Stream separation at Cape Hatteras. This is also done with models of the Gulf of Mexico to set the inflow conditions at the Yucatan Straits. This subroutine is not used in all models.

The initial bottom friction coefficient CBC(I,J) is calculated depending on the depth and grid spacing, but is at least set to the minimum value CBCMIN=0.0025.

The user can print out for inspection whatever initial values are desired. A description of the various subroutines for writing output is discussed in the section on utilities subroutines.

The value of the CFL criterion for each grid square, considering the dimensions of the grid and the ocean depth, is calculated and stored in the Temporary Storage Space array TPS(I,J). The value of the external mode time step DTE must be smaller than the minimum value in this array. Note that the array TPS is used to store other values in other parts of the program.

In some versions of the model the the top and bottom levels of the turbulence length scale are set equal to zero with the statements L(I,J,1)=0.0 and L(I,J,KB)=0.0. Recall that the array L is defined as a real variable. The statements

```
L(I,J,K)=Q2LB(I,J,K)/(Q2B(I,J,K)+SMALL)
KQ(I,J,K)=0.20*L(I,J,K)*SQRT(Q2B(I,J,K))
```

first initialize the turbulence length. Note that Q2B is twice the turbulent kinetic energy. The parameter SMALL=1.E-10 in the denominator ensures that there will be no division by zero, when using the initial value of Q2B that was set to zero. The second line initializes the vertical mixing coefficient for turbulence, which is defined as a real number.

The model uses the leapfrog scheme for integration in time. It is necessary to keep three levels for the variables that are integrated in time, and usually the letter B at the end of a variable name denotes the BACK (or n-1) time step, and the letter F at the end of a variable name denotes the FORWARD (or n+1) time step. However, in order to save memory, the variable UF is used to represent the (n+1) time level for T and Q2, and the variable VF is used to represent the (n+1) time level for S and Q2L. It is necessary to initialize the model by specifying the variables at two time levels, the back and center levels. It is sufficient to set the variable at the center time step equal to the variable at the back time step. This is what is done in the array calculations

```
UA(I,J)=UAB(I,J)
VA(I,J)=VAB(I,J)
EL(I,J)=ELB(I,J)
ETB(I,J)=ELB(I,J)
ET(I,J)=ETB(I,J)
ETF(I,J)=ET(I,J)
D(I,J)=H(I,J)+EL(I,J)
DT(I,J)=H(I,J)+ET(I,J)
Q2(I,J,K)=Q2B(I,J,K)
Q2L(I,J,K)=Q2LB(I,J,K)
T(I,J,K)=TB(I,J,K)
S(I,J,K)=SB(I,J,K)
U(I,J,K)=UB(I,J,K)
V(I,J,K)=VB(I,J,K).
```

If the model is initialized by reading in a restart file that has two time levels, then the above commands should not be used.

The subroutine DENS is called to calculate the density, and the mean density can be initialized to this value, if it has not been defined before, with the command, RMEAN(I,J,K) = RHO(I,J,K).

The subroutine BAROPG is called to obtain the baroclinic pressure gradient CALL BAROPG(DRHOX,DRHOY,TRNU,TRNV). The four parameters in the call statement are calculated in BAROPG and returned. The parameters DRHOX(I,J,K) and DRHOY(I,J,K) are the X and Y components of the internal baroclinic forcing term, resulting from density differences. The parameters TRNU(I,J) and TRNV(I,J) are the components of the vertically integrated pressure gradient forcing. The components DRHOX, DRHOY are used in the three dimensional calculations, and their vertical integrals TRNU and TRNV are used in the two dimensional barotropic calculations.

Part 2. The Integration

The numerical integration takes place in the loop DO 9000 IINT=1,IEND. The index IINT counts the internal mode (longer) time step DTI, making three dimensional (baroclinic mode) calculations. For every one of these internal mode time steps, the model performs ISPLIT external mode (barotropic or two dimensional) calculations, integrating with the external mode (shorter) time step DTE. (Recall that the integer ISPLIT is the ratio of the internal mode to external mode time steps.) This is done in the loop DO 8000 IEXT=1,ISPLIT.

We begin with a discussion of the internal mode time integration. The model first determines the variable TIME, giving its value in days and fractions of a day

```
TIME=DAYI*DTI*FLOAT(IINT)+TIME0.
```

Some versions of the model calculate the Julian date which is useful for modeling case studies with tides or specific wind forcing data sets

JDAY=DAYI*DTI*FLOAT(IINT)+JDAY0.

The value of RAMP = TIME/PERIOD is between zero and one, and is the fraction of the variable PERIOD (in days) that has elapsed since the start of the time integration. The value of PERIOD could be arbitrarily set to one day (PERIOD=1.) or longer as required to spin up a particular problem. After the time interval PERIOD has elapsed, the value of RAMP is held constant at unity, with the command, IF(RAMP.GT.1.0) RAMP=1.0. This value is used to spin up a forcing gradually and avoid exciting transient waves in the model. Be sure that RAMP is set equal to one if the model is initialized from a restart file.

Generally, all initial forcings to the model should be ramped. The vertically integrated pressure gradient forcing components are ramped where they appear in the barotropic equations of motion, with the terms RAMP*TRNU, and RAMP*TRNV, and if the model is forced by any inflow boundary conditions, they are also ramped in subroutine BCOND.

For the three dimensional calculations (MODE=3 or 4) the model calls subroutine BAROPG with the command

```
IF(MODE.NE.2) THEN
CALL BAROPG
ENDIF
```

to calculate the vertically integrated baroclinic pressure gradient TRNU, TRNV. Then the statement

TRNU(I,J)=TRNU(I,J)+ADVUU(I,J)-ADVUA(I,J)

adds the vertical integral of the horizontal dispersion terms ADVUU calculated in subroutine ADVU, and subtracts the barotropic horizontal advection ADVUA calculated in subroutine ADVAVE. This is made here for the internal mode calculations.

The model then calculates the horizontal kinematic viscosity coefficient AAM(I,J,K) at each level. For some applications it can be kept at a constant value (typically 500 – 2000 $m^2 s^{-1}$).

The vertically integrated kinematic viscosity is initialized to zero with the statement AAM2D(I,J)=0.0, and the statement

AAM2D(I,J)=AAM2D(I,J)+AAM(I,J,K)*DZ(K)

vertically integrates the horizontal kinematic viscosity.

The model will calculate an average of the external mode variables over the ISPLIT external mode time steps in the internal mode time step. The variable EGF(I,J) is this average of the external mode sea level EL(I,J). The variables UTF(I,J) and VTF(I,J) are these averages of the external mode velocities, multiplied by the depth D(I,J)=H(I,J)+EL(I,J). This averaging calculation is performed as the time integration takes place. Before the start of the external mode calculation, these averages must be initialized. This is done in the loops for

EGF(I,J)=EL(I,J)*ISPI

UTF(I,J)=UA(I,J)*(D(I,J)+D(I-1,J))*ISP2I.

Recall that ISPI is the (real value) number of external mode time steps in one internal mode time step. The real value ISP2I is half this value. The ISP2I value is used to incorporate the division by 2 when averaging the two values of D(I,J). The averaging at each external mode time step will take place at the end of the external mode calculation.

The External Mode

The loop DO 8000 IEXT=1,ISPLIT, performs the external mode time step integrations. There are ISPLIT external mode (short) time steps DTE in one internal mode (long) time step DTI. The loop DO 405 J=2,JMM1, calculates the vertically averaged velocity fluxes FLUXUA(I,J), FLUXVA(I,J) through the sides of a grid square centered on a depth point. The loop DO 410 J=2,JMM1, calculates the barotropic sea level ELF(I,J) from the continuity equation. The command CALL BCOND(1), applies the boundary conditions and masking to the barotropic sea level ELF(I,J).

The integer ISPADV is the frequency interval in external mode time steps at which the advection subroutine ADVAVE is called. The model is usually set with ISPADV=1, and this is recommended for strongly advective regimes, such as Gulf Stream models. The statement

IF(MOD(IEXT,ISPADV).EQ.0) CALL ADVAVE(ADVUA,ADVVA,MODE)

makes this determination and call. This subroutine returns ADVUA, ADVVA, the nonlinear velocity advections of the vertically integrated velocity.

The barotropic velocity equation for UAF(I,J) is calculated in the loop DO 420 J=2,JMM1, and evaluates the forcings of the Coriolis, sea level pressure gradient, vertically integrated baroclinic pressure gradient due to density differences, and wind stress: The Coriolis terms are multiplied by the array CURV42D(I,J). Terms multiplied by GRAV are the sea level pressure gradient. A weighted average over the three time steps is sometimes taken using the parameter ALPHA (typically, ALPHA=0.225). This is not absolutely necessary, but is done to increase the time step. The vertically integrated baroclinic pressure gradient terms are TRNU. The surface momentum flux (wind stress forcing) terms are WUSURF. The quadratic bottom friction terms WUBOT are calculated in subroutine ADVAVE. The array ARU(I,J) is the area of a grid square centered on the U(I,J) velocity point, and is necessary because the equations are cast in generalized coordinates. The loop DO 425 J=2,JMM1,for the velocity UAF(I,J) integrates the vertically averaged U velocity forward in time using the centered difference (leapfrog) time step.

The loops DO 430 and DO 435 perform the analogous integration of the barotropic V velocity equation. The command CALL BCOND(2), calls subroutine BCOND to apply the boundary conditions and masking to the barotropic velocities UAF and VAF.

A smooth value for ETF(I,J), the surface elevation used for the internal mode calculations, is calculated with the statement

IF(IEXT.LT.(ISPLIT-2)) GO TO 440

and the subsequent commands up to the 440 CONTINUE, based on the last three external mode time steps.

The model next tests for instabilities which might occur if, for example, DTE or DTI is too large. The loop DO 441, in effect, finds the largest absolute value of the elements in the velocity array VAF. If this is greater than some specified value (in this case 100.0 ms^{-1}), the model will GO TO 9001, print out arrays, and then STOP. This allows a normal exit to the program when nonlinear instabilities are causing the model to blow up, and the printed arrays can give some idea of what is happening. Other instability checks can be written based on sea level height.

It is necessary to filter the variables over the three time steps of the central differencing. Otherwise the solutions at the odd and even time steps will tend to diverge. The model performs this filtering at every time step, but a model could be written that does this at every several time steps if desired. The loop DO 500 J=1,JM does this time filtering and then resets the time sequence for the next external mode time step calculation.

The ongoing averaging of the external mode variables over the ISPLIT external mode time steps is performed in the loop DO 450 J=1,JM. The statement

IF(IEXT.EQ.ISPLIT) GO TO 8000

ensures that the last time step is not included in the averaging. This is because it is included as part of the initialization at the start of the external mode time step loop, in the loops DO 401 and DO 400.

The external mode time step is ended with the statement 8000 CONTINUE, and the internal mode calculations are resumed. The command

IF(MODE.EQ.2) GO TO 8200

ensures that if the model is being run in the barotropic mode, the three dimensional calculations are not performed. The command

IF(IINT.EQ.1.AND.TIME0.EQ.0.) GO TO 8200

ensures that if it is the first internal mode time step (IINT=1), the model jumps to statement 8200 CONTINUE and does not perform the three dimensional calculations, since the values of these initial variables were given in the initialization before the numerical integration.

The internal and external modes have different truncation errors, so that the vertical integrals of the internal mode velocities are not the exact values calculated for the barotropic velocities. The model now adjusts U(I,J,K) and V(I,J,K) such that their vertical averages are equal to UA(I,J) and VA(I,J). The temporary storage array TPS is set to zero to initialize the vertical integration of the velocity U(I,J,K), which is performed by the command

TPS(I,J)=TPS(I,J)+U(I,J,K)*DZ(K).

The loop DO 302 K=1,KBM1 that performs the calculation

U(I,J,K)= (U(I,J,K)-TPS(I,J)) + (UTB(I,J)+UTF(I,J))/(DT(I,J)+DT(I-1,J))

subtracts the vertically averaged velocity stored in TPS from U(I,J,K) to obtain the deviation from the mean. It then adds the average of UTB(I,J) and UTF(I,J) divided by the depth (recall that when UTF is calculated in the loop DO 450, it is defined as a velocity multiplied by the depth). This procedure ensures that the vertical average of U(I,J,K) is equal to UA(I,J). The three DO loops through statement numbers 303, 304, and 306 perform the analogous calculation for the V(I,J,K) velocity.

The vertical velocity is determined with the command CALL VERTVL(DTI2), and the command CALL BCOND(5) applies the boundary conditions (masking) to the vertical velocity.

The model now calculates twice the turbulent kinetic energy Q2(I,J,K) and the turbulence macroscale Q2L(I,J,K) at the forward time step. If the model followed the usual naming convention for the leapfrog time step, these arrays would be named Q2F and Q2LF. However, to save array space, these forward time step variables are stored temporarily in arrays UF and VF, respectively. First these arrays UF and VF are initialized to zero.

The statement CALL ADVQ(Q2B,Q2,DTI2,UF) has input parameters Q2B and Q2, twice the turbulent kinetic energy at the back and centered time steps, and twice the internal mode time step DTI2. The returned parameter is the advection for twice the turbulent kinetic energy at the forward time step, stored in array UF.

The statement CALL ADVQ(Q2LB,Q2L,DTI2,VF) has input parameters Q2LB and Q2L, for the back and centered time steps, and the time step DTI2. The returned parameter is the advection for variable at the forward time step, stored in array VF.

The command CALL PROFQ(DTI2) calls this subroutine to solve for the vertical profile of the turbulent kinetic energy and turbulence macroscale, still stored as UF and VF.

The command CALL BCOND(6) applies the boundary conditions (masking) to the turbulent kinetic energy and turbulence macroscale, still stored as UF and VF. Most of the model variables that are integrated in time are filtered with an Asselin (1972) filter to prevent the solutions at odd and even time steps from diverging. The turbulent Kinetic energy and turbulent length scale are filtered over the three time steps (Q2B,Q2,UF) and (Q2LB,Q2L,VF) using the commands

```
Q2=Q2+0.5*SMOTH*(UF+Q2B-2.0*Q2)
Q2L=Q2L+0.5*SMOTH*(VF+Q2LB-2.0*Q2L),
```

and the time sequence is reset with the commands

```
Q2B=Q2
Q2=UF
Q2LB=Q2L
Q2L=VF.
```

The model next computes the temperature and salinity. If the model followed the usual naming convention for the leapfrog time step, the variables would be named TF and SF. However, these variables will be stored in the arrays UF and VF to save space. The command

```
IF(MODE.EQ.4) GO TO 360
```

is used to skip over these calculations for temperature and salinity when the model is being run in the three dimensional diagnostic mode, which holds the temperature and salinity (and hence density) to their initial values. The command

```
CALL ADVT(TB,T,TMEAN,TSURF,DTI2,UF)
```

calls the horizontal scalar advection subroutine, with the input parameters of the temperature at the back (TB) and centered (T) time steps, the mean temperature (TMEAN), the surface temperature (TSURF), and twice the internal mode time step (DTI2). The subroutine uses the leapfrog time step to calculate the temperature at the forward time step, and stores it in array UF as the returned parameter. The command

```
CALL ADVT(SB,S,SMEAN,SSURF,DTI2,VF)
```

calls the advection subroutine with the input parameters of the salinity at the back (SB) and centered (S) time steps, the mean salinity (SMEAN), the surface salinity (SSURF), and twice the internal mode time step. The returned parameter is the salinity at the forward time step, stored in array VF. The commands

```
CALL PROFT(UF,WTSURF,DTI2)
CALL PROFT(VF,WSSURF,DTI2)
```

call this subroutine to determine the vertical profiles of scalar quantities. The input parameters are the surface temperature and salinity fluxes, WTSURF and WSSURF, the time step, and the temperatures and salinities stored in UF and VF. The vertical profiles of temperature and salinity are calculated and returned in UF and VF. The command CALL BCOND(4) calls the subroutine to apply the boundary conditions and land mask to the temperature and salinity, stored as UF and VF. The temperature and salinity are filtered over the three time steps using an Asselin (1972) filter with the commands

```
T=T+0.5*SMOTH*(UF+TB-2.0*T)
S=S+0.5*SMOTH*(VF+SB-2.0*S),
```

and the time sequence is reset with the commands

```
TB=T
T=UF
SB=S
S=VF.
```

The density is calculated with the statement CALL DENS. This marks the end of the

prognostic calculations for temperature, salinity, and density. These calculations would have been skipped in the diagnostic mode (MODE=4).

The last set of model prognostic calculations is for the velocities UF and VF. The commands

```
CALL ADVU(DRHOX,ADVUU,DTI2)
CALL ADVV(DRHOY,ADVVV,DTI2)
```

call the subroutines to calculate the horizontal changes for the U and V velocities. The calling parameters are the baroclinic pressure gradient terms (DRHOX, DRHOY), the vertical integral of the horizontal dispersion terms (ADVUU, ADVVV), and twice the internal mode time step DTI2. These subroutines include the influences of horizontal advection and diffusion, Coriolis, and baroclinic pressure gradient terms. The centered difference scheme is used to calculate UF and VF. The commands CALL PROFU(DTI2) and CALL PROFV(DTI2) call the subroutines to calculate the vertical profile of the internal mode velocities UF and VF. The command CALL BCOND(3) is used to apply the boundary conditions and velocity masks to UF and VF. These velocities are then filtered over the time steps with an Asselin (1972) filter.

The time series is reset with the commands

```
UB(I,J,K)=U(I,J,K)
U(I,J,K)=UF(I,J,K)
VB(I,J,K)=V(I,J,K)
V(I,J,K)=VF(I,J,K)

EGB(I,J)=EGF(I,J)
ETB(I,J)=ET(I,J)
ET(I,J)=ETF(I,J)
DT(I,J)=H(I,J)+ET(I,J)
UTB(I,J)=UTF(I,J)
VTB(I,J)=VTF(I,J).
```

This concludes all the time step calculations for the numerical integration of the internal mode time step. The command CALL FINDPSI is used to calculate the stream function PSI(I,J). The remaining commands are to print out or save information to various files at selected intervals, and the user will modify these to suit the application. For example, the command

```
IF(MOD(INT,IPRINT).NE.0) GO TO 7000
```

is used to print out various fields every IPRINT number of internal mode time steps IINT. The command

```
IF(ABS(VAMAX).GT.100.0.OR.ABS(VAMIN).GT.100.0) STOP
```

is used to stop the caclulation in case of numerical instabilities. The statement 9000 CONTINUE ends the internal mode time step DO loop.

For long time simulations of the model, it may be desirable to write model data to a restart file. It is best to write binary data files for greater precision. Since the model uses a centered difference (leapfrog) time step, two time levels are necessary for a 'seemless' restart. This can be done with the command

```
      WRITE(77) TIME,
     1  CURV42D,WUBOT,WVBOT,AAM2D,UA,UAB,VA,VAB,EL,ELB,
     2  ET,ETB,EGB,UTB,VTB,U,UB,W,V,VB,T,TB,S,SB,RHO,ADVUU,
     3  ADVVV,ADVUA,ADVVA,KM,KH,KQ,L,Q2,Q2B,AAM,Q2L,Q2LB.
```

5. THE COMPUTATIONAL MOLECULE

The model equations are finite differenced on an Arakawa-C staggered grid, shown in Fig. 1 for the external mode calculation. The ocean depths H(I,J) and sea level EL(I,J) are specified or calculated at the center point of a grid square. The east-west velocity UA(I,J) is calculated on the western side of the grid square, and the north- south velocity VA(I,J) is calculated on the south side of the grid square. For three dimensional calculations, U(I,J,K) is calculated on the west side, V(I,J,K) on the south side, and T(I,J,K), S(I,J,K), and RHO(I,J,K) at the center of the grid square at each intermediate level ZZ(K). The vertical velocity W(I,J,K), turbulent kinetic energy Q2(I,J,K), turbulent length scale L(I,J,K), their product Q2L(I,J,K), and diffusivities KM(I,J,K) and KH(I,J,K) are all calculated at the center point of a grid square, but at the Z(K) level in the vertical.

When a variable is calculated at a grid point from the finite difference equations, some variables at surrounding grid points are used in the calculation. This pattern of influence of the surrounding grids in the calculation of the particular variable is called the "computational molecule", and depends on the form of the finite difference equations. The computational molecule for the sea level, temperature, salinity, and other variables calculated at the center of a grid square is shown in Fig. 2. The computational molecules for the U and V velocities are shown in Figs. 3 and 4, respectively. These latter two molecules reach out farther to adjacent grid points because of the form of the momentum advection and diffusion terms.

The calculations are carried out over the entire model domain, over both land and ocean points. This is why land points are set to a small nonzero value (typically less than one meter), so that division by zero does not occur when dividing by D(I,J)=H(I,J)+EL(I,J) in the calculations. When calculations are performed at the boundary gridpoints, or interior points close to the boundary, the computational molecule reaches out to values outside the domain, and the bounds of some arrays may be exceeded. The user should check the compiler options for the particular computer being used, to see that this is allowed. When this situation occurs, the computer selects some value in an adjacent array stored in the COMMON block memory location. This is why all variables in the COMMON block are initialized to zero or some value at the start of the program and redefined later: to avoid an "undefined variable" error message if a calculation exceeds the bounds of an array. When a calculation makes use of values from outside the bounds of an array, an erroneous value is produced. The program must then replace this value before subsequent calculations are made, to avoid this error propagating inward from the boundaries. If land points are at the edge of the domain this is done by multiplying by the appropriate land or velocity mask FSM(I,J), DUM(I,J), or DVM(I,J). If open boundaries are at the edge of the domain, the open boundary conditions must overwrite any spurious values.

The model equations are linearized near the boundaries to prevent the computational molecule there from extending outside the domain. Also, for many applications, it is desirable to apply linear open boundary conditions. At the eastern and western boundaries, this is done by setting

```
ADVUA(1,J)=0
ADVUA(IM,J)=0
ADVVA(1,J)=0
ADVVA(IM,J)=0
```

in subroutine ADVAVE. At the northern and southern boundaries the computational molecule does not extend outside the domain because the variables are calculated at J=2,...,JMM1, or J=3,...,JMM1 (where JMM1 = JM–1) in the MAIN program and subroutines. The values at the boundary are then specified explicitly as open boundary conditions at J=1 and J=JM (and sometimes at J=2 and J=JMM1 depending on the application). Fig. 1 is a schematic of the Arakawa-C grid and indicates which variables must be calculated linearly at the boundary, and which variables must be specified explicitly as open boundary conditions.

The model is set up with the assumption that the western boundary is a wall, and should run without difficulty in its present form when appropriate boundary conditions are specified. Should the user change the model equations, such as by adding a new diffusion algorithm that changes the computaional molecule, or by configuring the model to a new domain, it is his responsibility to ensure that any spurious values resulting from computations near the boundaries are overwritten by the masks or appropriate boundary conditions.

6. SUBROUTINE BCOND

The boundary conditions are applied in subroutine BCOND. Basically, this subroutine is divided into six independent sections that apply the appropriate boundary conditions to the variables. This subroutine is called from the MAIN program with the command

```
CALL BCOND(IDX)
```

where the calling parameter IDX is an integer from 1 to 6. Then at the start of subroutine BCOND the computed GO TO branch statement

```
GO TO (10,20,30,40,50,60), IDX
```

uses this value of IDX to direct the program to the appropriate section where the boundary conditions are applied. After the specification of any open boundary conditions, the variables at land regions are set to zero by multiplying by the appropriate land FSM(I,J) or velocity DUM(I,J), DVM(I,J) mask, and control is returned to the MAIN program with a RETURN statement. The masking is done after the open boundary conditions are applied, because they will then overwrite any specifications made at a land boundrary. In general, it is not necessary to specify boundary conditions at a land boundary, since the masking will automatically set to zero the normal component of velocity there.

For some problems such as a hurricane passage through the Gulf of Mexico, the wind stress and atmospheric pressure must be specified as a function of time. This can be done conveniently by adding a section to subroutine BCOND for the atmospheric forcing. At present the subroutine BCOND has six sections for the boundary conditions on the model variables.

The Sea Level

For IDX=1, boundary conditions are applied to the external mode sea level ELF(I,J). For some applications, such as when the open boundary is along a shelf break, the sea level can be set to zero. In many cases the sea level is set equal to the upstream value, especially at outflow boundaries. It is often useful to apply radiation conditions (Chapman 1985) at cross shelf boundaries. If it is desired to explicitly give values at a north, east, or south boundary, then boundary arrays ELN(IM), ELE(JM), or ELS(IM) can be specified, and the value of ELF(I,J) set equal to these values at the boundary. The sea levels are masked with the land mask ELF(I,J)=ELF(I,J)*FSM(I,J).

The External Mode Velocities

For IDX=2, boundary conditions are applied to the external mode velocities UAF(I,J) and VAF(I,J). For some applications such as modeling the Gulf Stream or Gulf of Mexico, the inflow (and sometimes the outflow) velocities can be specified. This can be done by specifying the boundary arrays UABE(JM), VABN(IM), and VABS(IM). At outflow boundaries, the normal component of velocity can be set to the value immediately upstream. For some applications, radiation conditions can be used. The velocities are masked by the arrays

```
UAF(I,J)=UAF(I,J)*DUM(I,J)
```

VAF(I,J)=VAF(I,J)*DVM(I,J).

At a land boundary, the masking by DUM(I,J) and DVM(I,J) automatically applies the condition that the normal component of velocity vanishes there.

The Internal Mode Velocities

For IDX=3, boundary conditions are applied to the internal mode velocities UF(I,J,K) and VF(I,J,K). Usually they are radiation conditions or upstream conditions. For some applications, the inflow velocity could be specified as a function of depth. The velocities are masked with the arrays

UF(I,J,K)=UF(I,J,K)*DUM(I,J)

VF(I,J,K)=VF(I,J,K)*DVM(I,J).

The masking automatically applies the condition that the normal component of velocity vanishes at the boundary.

The Temperature and Salinity

For IDX=4, boundary conditions are applied to the temperature TF(I,J,K) and salinity SF(I,J,K), which are being stored in the arrays UF(I,J,K) and VF(I,J,K), respectively.

The temperature can be specified on the boundaries with the boundary value arrays TBN(IM,KB), TBE(JM,KB), and TBS(IM,KB). These arrays can be set to the appropriate values of climatology, with 10 C^o subtracted. At the boundary the value of TF(I,J,K), stored in UF(I,J,K), is set equal to the boundary value array.

The salinity can be specified on the boundaries with the boundary value arrays SBN(IM,KB), SBE(JM,KB), and SBS(IM,KB). These arrays can be set to the appropriate values of climatology with 35 ppt subtracted. At the boundary the value of SF(I,J,K), stored in VF(I,J,K), is set equal to the boundary value arrays. The temperatures and salinities are often held to a constant value at an inflow boundary, and set to upstream values and an outflow boundary. Both temperature and salinity are masked with the land mask FSM.

The Vertical Velocity

For IDX=5, boundary conditions are applied to the sigma coordinate vertical velocity W(I,J,K). The vertical velocity is masked with the land mask FSM(I,J). For some applications, the vertical velocity on the eastern boundary W(IM,J,K) is set to zero.

The Turbulent Kinetic Energy

For IDX=6, boundary conditions are applied to the turbulent kinetic energy Q2F(I,J,K) stored in UF(I,J,K), and twice the turbulent kinetic energy times the turbulence length scale Q2LF(I,J,K) stored as VF(I,J,K). They are masked with the land mask FSM(I,J). For some applications, these values are set equal to a constant at an inflow boundary.

The model has several subroutines for printing out data fields for ready visualization. Their use is described briefly at the beginning of each print subroutine. Here we describe their use in more detail. It should be kept in mind that some earlier versions of the model may have different, but similar subroutines for printing output. Here we describe the subroutines used in the Princeton model pmod.f.

SUBROUTINE PRXY

This subroutine prints two dimensional (X,Y) arrays, and is probably the most frequently used print routine. The calling parameters are given

SUBROUTINE PRXY(LABEL,TIME,ARRAY,IM,ISKIP,JM,JSKIP,SCALA)

LABEL	The name of the field to appear on the printout, passed in quotes as a character string.
TIME	The variable TIME, giving elapsed time in days. At initialization, be sure to set TIME=0.0 to use these subroutines.
ARRAY	The variable to be printed, of dimensions (IM,JM), eg., H for the ocean depths, EL for the surface elevation.
IM	The first dimension of the array in the X–direction.
ISKIP	The frequency at which variables in the X–direction are skipped when printing. To print every third value, set ISKIP=3; to print every other value, set ISKIP=2; to print every value, set ISKIP=1.
JM	The second dimension of the array in the Y–direction.
JSKIP	The frequency at which variables in the Y–direction are skipped when printing. To print every third value, set JSKIP=3; to print every other value, set JSKIP=2, to print every value, set JSKIP=1.
SCALA	The scale used to print values. The subroutine prints out the actual values divided by the scale. The subroutines also print out the message "MULTIPLY BY" and the value of the scale. If SCALA is zero, the scale is computed by the subroutine, which is useful if the magnitude of the variable is unknown.

The use of the scale is illustrated here with several useful examples. To display the grid array on the order of several tens of thousands of meters, say DX(I,J)=10000.0, set SCALA=1.E+3, and the PRXY subroutine will print out an array of numbers, each of value 10, corresponding to kilometers.

To display currents which are typically less than one meter per second, say UAB(I,J)=0.30, set SCALA=1.E–2, and the PRXY subroutine will print out an array of numbers each of value 30, corresponding to centimeters per second.

To display sea levels that are typically less than one meter, say ELB(I,J)=0.05, set SCALA=1.E–2, and PRXY prints out an array of numbers each with value 5, corresponding to sea level in centimeters.

To display the wind stress (divided by the water density) in units of $m^2 s^{-2}$, say WUSURF(I,J) = 2.0E–4, set SCALA=1.E–4, and PRXY prints out an array of numbers each of value 2, corresponding to wind stress in $dynes\, cm^{-2}$ (divided by the water density $1.0\, gm\, cm^{-3}$).

The SCALA is set similarly in the other print subroutines below.

SUBROUTINE PRXYZ

This subroutine prints horizontal (X–Y) slices of a three dimensional array. The calling parameters, besides those discussed above, are

SUBROUTINE PRXYZ(LABEL,TIME,ARRAY,IM,ISKIP,JM,JSKIP,KB,SCALA).

ARRAY	A three dimensional array, of dimensions (IM,JM,KB), eg., T for temperatures, S for salinities.
KB	The dimension in the vertical.

The subroutine internally choses the X–Y slices to print. This is done with the statements of the form

```
DIMENSION KP(3)
DATA KE,KP/3,1,3,5/
```

which in this case dimensions an array KP(KE) and specifies that the levels to be printed will be K=1,3,5. To print out five levels, K=1,5,7,9,11, the user could modify the subroutine with the statements

```
DIMENSION KP(5)
DATA KE,KP/5,1,5,7,9,11/.
```

SUBROUTINE PRXZ

This subroutine prints vertical (X–Z) slices of a three dimensional array, for the slices at J=J1, J2. The output is also displayed with the number K of the layer and the value ZZ(K) printed in columns to the left. The value of the depth is printed as a row across the top. The calling parameters, besides those discussed above are

SUBROUTINE PRXZ(LABEL,TIME,ARRAY,IM,ISKIP,JM,J1,J2,KB,SCALA,DT,ZZ).

J1,J2	The Y values (at J=J1,J2) at which the the slice in the X–Z plane are taken.
DT	The depth of the water column, array DT(IM,JM).
ZZ	The array specifying the midpoints of the levels ZZ(KB).

SUBROUTINE PRYZ

This subroutine prints vertical (Y–Z) slices of a three dimensional array. This version of the subroutine prints the slices at I=IM/2 and I=IM–3. In this subroutine ISKIP is used only in the determination of the scale if SCALA=0. The subroutine also prints out the value K of the vertical layer and corresponding value ZZ(K) to the left of the output, and the depth in meters in a row across the top. The calling parameters are

SUBROUTINE PRYZ(LABEL,TIME,ARRAY,IM,ISKIP,JM,JSKIP,KB,SCALA,DT,ZZ).

As a final note, these subroutines internally make use of the one dimensional arrays IDT, JDT, LINE, and NOM to set up the printing layout on the pages. These arrays must be dimensioned to the same value which is greater than both IM and JM, or an error message will result. It is usually best to dimension these arrays to some high value, say IDT(500), JDT(500), LINE(500), and NOM(500) to prevent the necessity of frequent changes.

8. BOX MODEL TEST CASE

This is a box model test case to demonstrate the application of the Princeton model. It is a rectangular domain of dimension 2000 km in the east – west direction, 1000 km in the north – south direction, and of constant depth 2000 m. A simple sinusoidal wind stress profile is set up over the basin with easterly winds over the southern part and westerly winds over the northern part, that remain constant in time. The model has $20 \times 20 \times 24 = 9600$ total node points so that it can be run on most computers. The specifications are listed below.

(1) The box model dimensions are

IM = 20, JM = 20, KB = 24.

(2) A rectangular cartesian domain is used with grid spacing

DX(I,J) = 100.E3 (100 Km east – west)

DY(I,J) = 50.E3 (50 Km north – south).

(3) The coriolis parameter varies linearly with a beta plane approximation

BETA = 1.98E–11

COR4(I,J)=0.25*(7.292E–5+BETA*(J–1.0)*DY(I,J)).

(4) A constant depth of 2000 m is specified so that H(I,J) = 2000. To indicate a land boundary (the walls of the box model) we set

H(I,1) = H(I,JM) = 0.5 for I=1,...,IM

H(1,J) = H(IM,J) = 0.5 for J=1,...,JM.

(5) The temperature is specified with the exponentialy decreasing profile, with the first six levels at a constant value

TB(I,J,K)=10.0*EXP(ZZ(K))+10.0

IF(K.LT.6) TB(I,J,K)=TB(I,J,6).

The salinity is specified to be a constant everywhere SB(I,J,K)=35. It is necessary to subtract 10 degrees Celsius from the temperature and 35 parts per thousand from the salinity as part of the initialization. These values are added back when computing the density in SUBROUTINE DENS.

TB(I,J,K)=TB(I,J,K)–10.

```
SB(I,J,K)=SB(I,J,K)-35.
```

A call is made to subroutine DENS to determine the density RHO(I,J,K), and for this example, the value of the mean density array is set equal to RMEAN(I,J,K) = RHO(10,5,K). The mean values of temperature and salinity are set equal to TMEAN(I,J,K) = TB(I,J,K), and SMEAN(I,J,K) = SB(I,J,K).

(6) For an enclosed domain, it is not necessary to specify any boundary value arrays. The requirement that the normal component of velocity must vanish on a boundary is applied with the DUM(I,J) and DVM(I,J) velocity masks in SUBROUTINE BCOND, and this will overwrite any specifications using the boundary value arrays.

(7) The momentum fluxes WUSURF(I,J) and WVSURF(I,J) are the negative of the wind stress, divided by the density of water, expressed in the MKS units of $m^2 s^{-2}$. They are typically of magnitude 1.E–4. This example uses a simple sinusoidal profile with momentum input from easterly winds over the southern part of the domain, and from westerly winds over the northern part.

```
      WUSURF(I,J)=(1.E-4*COS(PI*(J-1)/JMM1))
     1  *0.25*(DVM(I,J+1)+DVM(I-1,J+1)+DVM(I-1,J)+DVM(I,J))
      WVSURF(I,J)=0.
```

The average over the velocity mask reduces the value of the wind stress at the boundary, to reduce boundary effects.

(8) The surface temperature and salinity fluxes will be set to zero with the commands WTSURF(I,J)=0, and WSSURF(I,J)=0.

(9) The model run will use an external model time step (DTE) of 192 s, and an internal mode time step (DTI) of 5760 s. This is set in the model by the commands DTI=5760 and ISPLIT=30.

(10) The initial time is set to zero with the command TIME=0. The model will be set to run 5 days with the command IDAYS=5, and the print intervals will be specified with this same value, so that there will be only one printout to the standard out (UNIT=6) at the end of the model run IPRTD1=5 and IPRTD2=5. In this application the value of ISWITCH is not used. The data can be written to files to be saved as desired (see User Supplied Input).

(11) The three dimensional calculation will be made by setting the parameter MODE=3.

(12) The vertical kinematic viscosity KM(I,J,K), vertical heat diffusivity KH(I,J,K), turbulent kinetic energy Q2B(I,J,K), and turbulence macroscale Q2LB(I,J,K) are initialized to zero or some small value

```
Q2B(I,J,K)=0

Q2LB(I,J,K)=0
```

KM(I,J,K)=2.E–4

KM(I,J,1)=KM(I,J,KB)=0

KH(I,J,K)=KM(I,J,K)

and the horizontal kinematic viscosity is initialized to the value AAM(I,J,K)=50.0.

The model is initialized from a state of rest so that the initial velocities and sea level are zero. This is taken care of in the initialization at the start of the MAIN program, where all values are set to zero before the values for each application are specified or read in from an input file.

(13) The PERIOD of time in days over which the calculations are spun up, or RAMP–ed, is one inertial period, that value of the inertial period at the center of the domain

PERIOD = DAYI*(2.0*PI)/(COR4(IM/2,JM/2)*4.0).

The other parameters in the model can usually be kept at the values originally specified, e.g.,

ISPADV=1

TPRNU=1.0

SMOTH=0.10

HORCON=0.02.

(14) Here we display the model parameters and model output, making use of the utilities print subroutines. First we wish to see the vertical levels and spacing. This is determined in subroutine DEPTH and printed from that subroutine. The calling parameters are KB and KBM1 and are given in the call statement

CALL DEPTH(Z,ZZ,DZ,DZZ,DZR,KB,KBM1).

Some earlier versions of subroutine DEPTH may have a different parameter list, but the Z, ZZ, DZ, and DZZ are always determined. The model output that is printed to the standard out (UNIT=6) is shown in Fig. 5.

The constant model depths of 2000 m in array H(I,J) can be displayed with the call

CALL PRXY('TOPOGRAPHY',TIME,H,IM,2,JM,1,1.E0) .

The array will be printed with every other value of I omitted, and the results are shown in Fig. 6. Since the scale is 1.E0 the values 2000 are printed. The land values along the boundary were set to the value 0.5 m, and so are printed as 0 when this scale is used. In

general, the variables over land points are set to zero when multiplied by the land mask in subroutine BCOND.

The CFL stability criterion time step for the model was calculated and stored in the temporary storage space array TPS(I,J) and can be displayed with the call

CALL PRXY('CFL TIME STEP',TIME,TPS,IM,2,JM,1,1.E0).

The values in the array are the constant 159 s, since the depths and grid spacings were held constant (Fig. 7). However, the CFL calculation is useful when setting up a model for any given problem.

The Coriolis parameter which varies with the latitude of each grid box can be displayed with the call statement

CALL PRXY('COR',TIME,COR,IM,2,JM,1,1.E–8).

A typical value of 8082 printed out in Fig. 8 means that the value of the Coriolis parameter is $8082 \times 10^{-8}\ s^{-1}$ since the scale is 1.E–8. Some versions of the model use the variable COR4(IM,JM) which is the value of the Coriolis parameter divided by 4.

The vertical distribution of initial temperature can be displayed with the call statement

CALL PRXZ('TEMP.',TIME,TB,IM,2,JM,5,10,KB,1.E–1,DT,ZZ).

which prints out two cross sections in the X–Z plane at the rows J=5, 10. Since these are identical because of the uniform initial conditions, only one slice is shown in Fig. 9. Since the SCALE=1.E–1, a typical value of 88.2 printed out means that the model temperature is $88.2 \times 10^{-1}\ C^o$ or $8.82\ C^o$. Since 10^o was subtracted from the temperature as part of the initialization, this represents a true temperature of $18.82\ C^o$. Note that the bottom layer K=KB always has zero values.

The initial density distribution can be displayed with the call statement

CALL PRYZ('RHO',TIME,RHO,IM,2,JM,2,KB,1.E–4,DT,ZZ).

This subroutine prints out two cross sections in the Y–Z plane, at I=IM/2=10 and I=IM–3=17. (The value of ISKP=2 is not used when the scale is specified as nonzero.) Because the cross sections are identical only one is shown here in Fig. 10. Since the scale is 1.E–4, a value in the printout of 1.4 means that the model value in array RHO is 1.4×10^{-4}. Since the model uses a normalized density for greater precision when calculating baroclinic pressure gradients, the actual physical density is found by adding 1.025 and multiplying the result by 1000, or

$$(1.025 + 0.00014) \times 1000 = 1025.14\ Kg\ m^{-3}.$$

Note that the bottom layer K=KB always has zero values.

The model output after the five day run can be examined. The surface momentum flux in the east–west direction driving the model can be displayed with the call statement

CALL PRXY('WUSURF',TIME,WUSURF,IM,2,JM,1,1.0E–8)

and the results are shown in Fig. 11. Since the surface momentum flux input WUSURF has

the opposite sign as the wind stress, the momentum input from the easterly winds over the southern part of the domain is positive, while the momentum input from the westerly winds over the northern part is negative. Since the scale is 1.0E–8, a typical value of 8794 means that the wind stress divided by the water density is $8794 \times 10^{-8}\ m^2 s^{-2}$ resulting from a wind stress of magnitude 0.8794 dynes cm^{-2}.

The sea level of the gyre that is set up by the forcing can be displayed by the call statement

```
CALL PRXY('FREE SURFACE',TIME,ELB,IM,2,JM,1,1.E–4).
```

The print out is shown in Fig. 12. Since the scale is 1.E–4, a value of 230 means that the sea level is $230 \times 10^{-4}\ m$, or 2.3 cm.

The current structure of this gyre can be displayed by the calls

```
CALL PRXY('AVERAGE U COMP.',TIME,UAB,IM,2,JM,1,1.E–4)
CALL PRXY('AVERAGE V COMP.',TIME,VAB,IM,2,JM,1,1.E–4)
```

and this is shown in Figs. 13 and 14. The scale is 1.E–4 so that a printed value of 124 means that the current velocity is $124 \times 10^{-4}\ ms^{-1}$, or 1.24 $cm\ s^{-1}$.

REFERENCES

Asselin, R. (1972) Frequency Filters for Time Integrations, *Mon. Weather Rev.*, **100**, 487–490.

Blumberg, A.F. (1977) Numerical Tidal Model of Chesapeake Bay, *Journal of the Hydraulics Division, ASCE*, **103**, 1–10.

Blumberg, A.F., and H.J. Herring (1987) Circulation Modelling Using Orthogonal Curvilinear Coordinates, In: Nihoul, J.C.J. and Jamart, B.M. (editors), *Three–Dimensional Models of Marine and Estaurine Dynamics*, Elsevier, New York, pp. 55–88.

Blumberg, A.F., and L.H. Kantha (1985) Open Boundary Condition for Circulation Models, *Journal of Hydraulic Engineering, ASCE*, **111**, 237–255.

Blumberg, A.F., and G.L. Mellor (1983) Diagnostic and Prognostic Numerical Circulation Studies of the South Atlantic Bight, *Journal of Geophysical Research*, **88** (C8), 4579–4592.

Blumberg, A.F., and G.L. Mellor (1985) A Simulation of the Circulation in the Gulf of Mexico, *Israel Journal of Earth Sciences*, **34**, 122–144.

Blumberg, A. F., and G.L. Mellor (1987) A Description of a Three–Dimensional Coastal Ocean Circulation Model, In: N.S. Heaps (Editor), *Three Dimensional Coastal Ocean Models, Coastal and Estuarine Sciences*, **4**, American Geophysical Union, Washington, D.C., pp. 1–16.

Blumberg, A.F., and L.–Y. Oey (1985) Modeling Circulation and Mixing in Estuaries and Coastal Oceans, In: S. Manabe (Editor), Issues in Atmospheric and Oceanic Modeling, Part A, Climate Dynamics, *Advances in Geophysics*, **28**, 525–547.

Chapman, D.C. (1985) Numerical Treatment of Cross–Shelf Open Boundaries in a Barotropic Coastal Ocean Model, *Journal of Physical Oceanography*, **15**, 1060–1075.

Galperin, B., and G.L. Mellor (1990a) A Time–Dependent, Three–Dimensional Model of the Delaware Bay and River System. Part 1: Description of the Model and Tidal Analysis, *Estaurine, Coastal and Shelf Science*, **31**, 231–253.

Galperin, B., and G.L. Mellor (1990b) A Time–Dependent, Three–Dimensional Model of the Delaware Bay and River System. Part 2: Three–Dimensional Flow Fields and Residual Circulation, *Estaurine, Coastal and Shelf Science*, **31**, 255–281.

Mellor, G.L. (1990) *Documentation for A Three–Dimensional, Primitive Equation, Numerical Ocean Model*, 41 pp., available from: Program in the Atmospheric and Oceanic Sciences, PO Box CN 710, Sayer Hall, Princeton University, Princeton, New Jersey, 08544–0710.

Mellor, G.L., and A.F. Blumberg (1985) Modeling Vertical and Horizontal Diffusivities with the Sigma Coordinate System, *Mon. Weather Rev.*, **113**, 1379–1383.

Mellor, G.L., and T. Ezer (1991) A Gulf Stream Model and an Altimetric Assimilation Scheme, *J. Geophys. Res.*, **96** (C5), 8779–8795.

Oey, L.–Y., G.L. Mellor, and R.I. Hires (1985a) A Three–Dimensional Simulation of the Hudson–Raritan Estuary. Part I: Description of the Model and Model Simulations, *J. Phys. Oceanog.*, **15**, 1676–1692.

Oey, L.–Y., G.L. Mellor, and R.I. Hires (1985b) A Three–Dimensional Simulation of the Hudson–Raritan Estuary. Part II: Comparisons with Observation, *J. Phys. Oceanog.*, **15**, 1693–1709.

Oey, L.–Y., G.L. Mellor, and R.I. Hires (1985c) A Three–Dimensional Simulation of the Hudson–Raritan Estuary. Part III: Salt Flux Analyses, *J. Phys. Oceanog.*, **15**, 1711–1720.

Shapiro, R. (1970) Smoothing, Filtering, and Boundary Effects, *Reviews of Geophysics and Space Physics*, 8 (2), 359–387.

LIST OF FIGURES

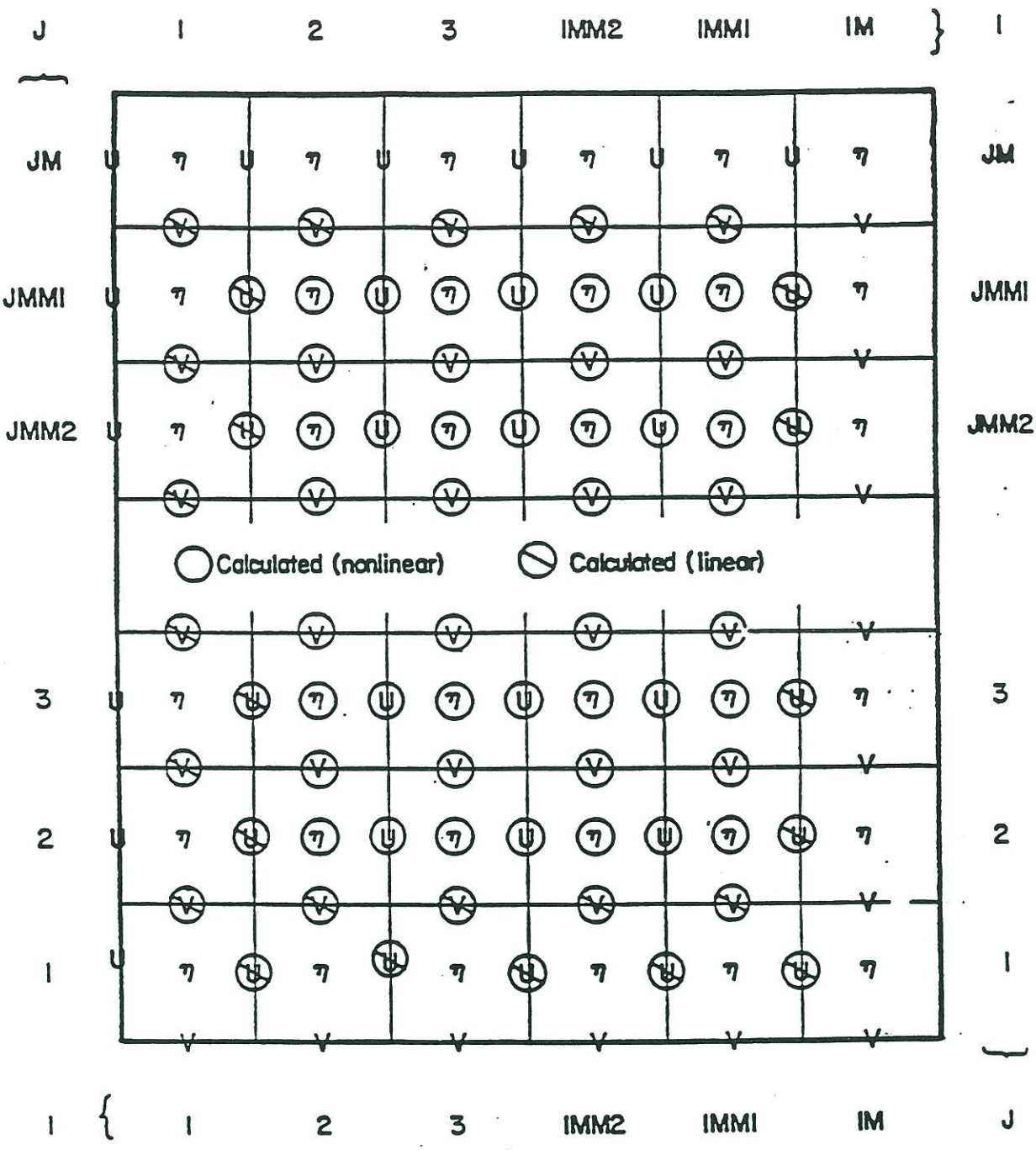


Fig. 1 The Arakawa–C staggard grid.

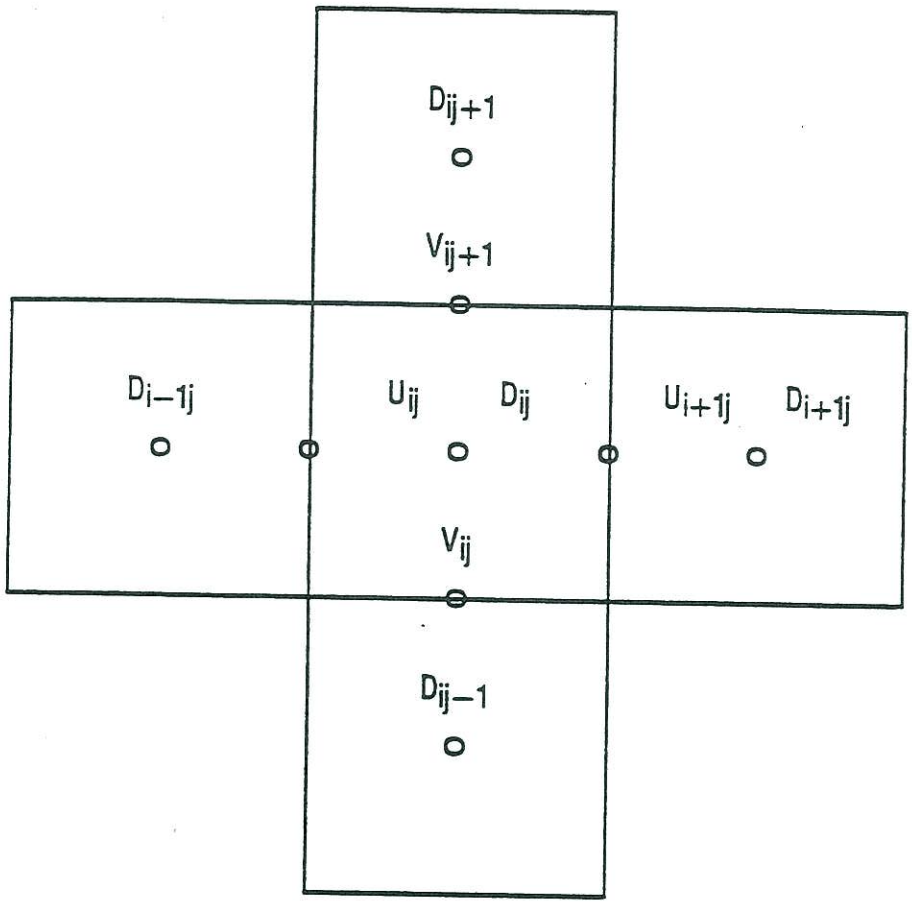


Fig. 2 The computational molecule for sea level, temperature, and other scalar variables.

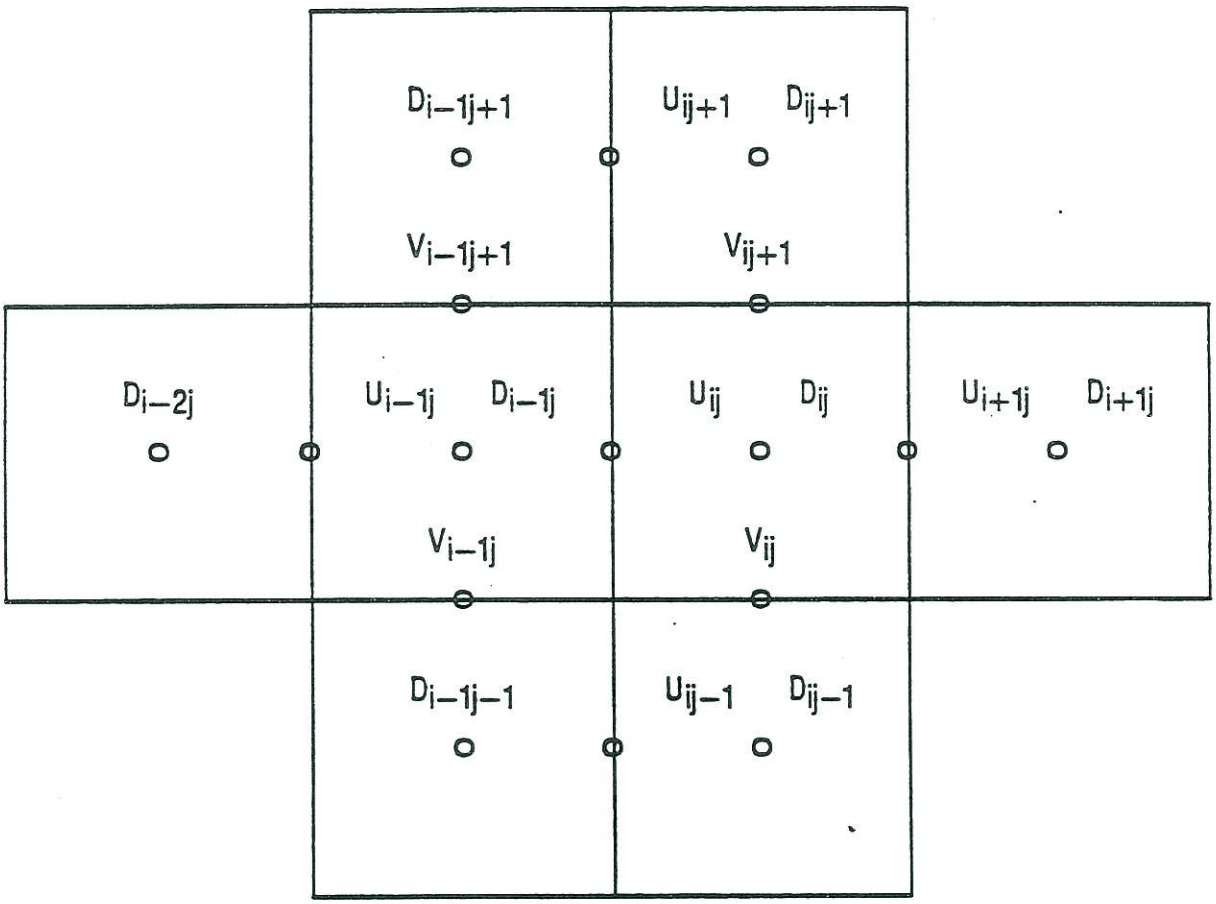


Fig. 3 The computational molecule for the U velocity.

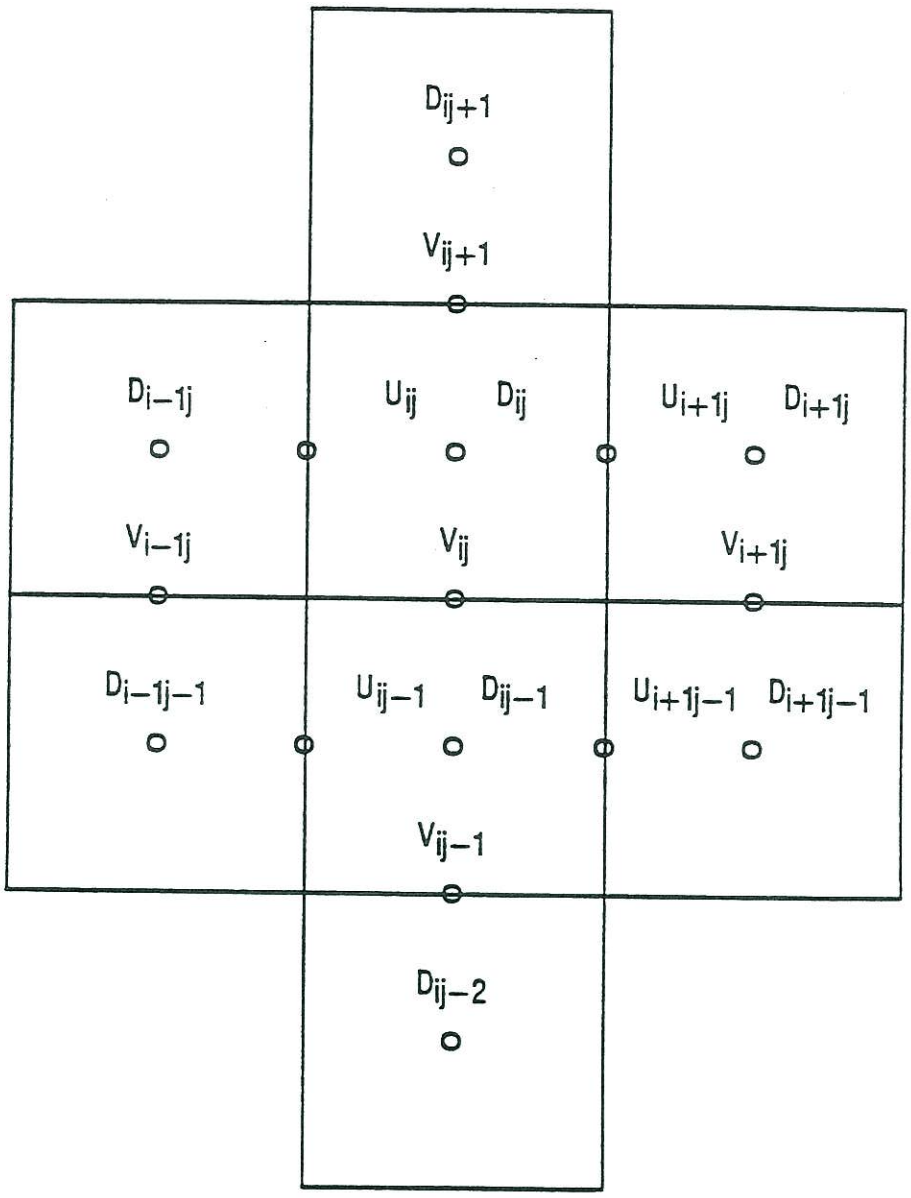


Fig. 4 The computational molecule for the V velocity.

K	Z	ZZ	DZ	DZZ
1	0.000	-0.003	0.006	0.006
2	-0.006	-0.009	0.006	0.009
3	-0.013	-0.018	0.013	0.018
4	-0.025	-0.035	0.025	0.035
5	-0.050	-0.071	0.050	0.054
6	-0.100	-0.125	0.050	0.050
7	-0.150	-0.175	0.050	0.050
8	-0.200	-0.225	0.050	0.050
9	-0.250	-0.275	0.050	0.050
10	-0.300	-0.325	0.050	0.050
11	-0.350	-0.375	0.050	0.050
12	-0.400	-0.425	0.050	0.050
13	-0.450	-0.475	0.050	0.050
14	-0.500	-0.525	0.050	0.050
15	-0.550	-0.575	0.050	0.050
16	-0.600	-0.625	0.050	0.050
17	-0.650	-0.675	0.050	0.050
18	-0.700	-0.725	0.050	0.050
19	-0.750	-0.775	0.050	0.050
20	-0.800	-0.825	0.050	0.050
21	-0.850	-0.875	0.050	0.054
22	-0.900	-0.929	0.050	0.046
23	-0.950	-0.975	0.050	0.050
24	-1.000	-1.025	0.000	0.000

Fig. 5 The box model vertical distribution of σ levels produced by a call to subroutine DEPTH.

TOPOGRAPHY

TIME = 0.0000 DAYS MULTIPLY ALL VALUES BY 1.000E+00

	1	3	5	7	9	11	13	15	17	19
20	1	1	1	1	1	1	1	1	1	1
19	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
18	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
17	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
16	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
15	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
14	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
13	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
12	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
11	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
10	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
9	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
8	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
7	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
6	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
5	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
4	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
3	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
2	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
1	1	1	1	1	1	1	1	1	1	1

Fig. 6 The box model 2000 m bathymetry, displayed by CALL PRXY('TOPOGRAPHY',TIME,H,IM,2,JM,1,1,E0).

CFL TIME STEP

TIME = 0.0000 DAYS MULTIPLY ALL VALUES BY 1.000E+00

	1	3	5	7	9	11	13	15	17	19
20	0	0	0	0	0	0	0	0	0	0
19	0	159	159	159	159	159	159	159	159	159
18	0	159	159	159	159	159	159	159	159	159
17	0	159	159	159	159	159	159	159	159	159
16	0	159	159	159	159	159	159	159	159	159
15	0	159	159	159	159	159	159	159	159	159
14	0	159	159	159	159	159	159	159	159	159
13	0	159	159	159	159	159	159	159	159	159
12	0	159	159	159	159	159	159	159	159	159
11	0	159	159	159	159	159	159	159	159	159
10	0	159	159	159	159	159	159	159	159	159
9	0	159	159	159	159	159	159	159	159	159
8	0	159	159	159	159	159	159	159	159	159
7	0	159	159	159	159	159	159	159	159	159
6	0	159	159	159	159	159	159	159	159	159
5	0	159	159	159	159	159	159	159	159	159
4	0	159	159	159	159	159	159	159	159	159
3	0	159	159	159	159	159	159	159	159	159
2	0	159	159	159	159	159	159	159	159	159
1	0	0	0	0	0	0	0	0	0	0

Fig. 7 The CFL criterion of 159 s for each grid square of the box model, displayed by CALL PRXY('CFL TIME STEP',TIME,TPS,IM,2,JM,1,1.E0).

COR

TIME = 0.0000 DAYS MULTIPLY ALL VALUES BY 1.000E-08

	1	3	5	7	9	11	13	15	17	19
20	9171	9171	9171	9171	9171	9171	9171	9171	9171	9171
19	9072	9072	9072	9072	9072	9072	9072	9072	9072	9072
18	8973	8973	8973	8973	8973	8973	8973	8973	8973	8973
17	8874	8874	8874	8874	8874	8874	8874	8874	8874	8874
16	8775	8775	8775	8775	8775	8775	8775	8775	8775	8775
15	8676	8676	8676	8676	8676	8676	8676	8676	8676	8676
14	8577	8577	8577	8577	8577	8577	8577	8577	8577	8577
13	8478	8478	8478	8478	8478	8478	8478	8478	8478	8478
12	8379	8379	8379	8379	8379	8379	8379	8379	8379	8379
11	8280	8280	8280	8280	8280	8280	8280	8280	8280	8280
10	8181	8181	8181	8181	8181	8181	8181	8181	8181	8181
9	8082	8082	8082	8082	8082	8082	8082	8082	8082	8082
8	7983	7983	7983	7983	7983	7983	7983	7983	7983	7983
7	7884	7884	7884	7884	7884	7884	7884	7884	7884	7884
6	7785	7785	7785	7785	7785	7785	7785	7785	7785	7785
5	7686	7686	7686	7686	7686	7686	7686	7686	7686	7686
4	7587	7587	7587	7587	7587	7587	7587	7587	7587	7587
3	7488	7488	7488	7488	7488	7488	7488	7488	7488	7488
2	7389	7389	7389	7389	7389	7389	7389	7389	7389	7389
1	7290	7290	7290	7290	7290	7290	7290	7290	7290	7290

Fig. 8 The Coriolis parameter in s^{-1} for each grid square of the box model, displayed by CALL PRXY('COR',TIME,COR,IM,2,JM,1,1.E-8).

TEMP.

TIME = 0.00 MULTIPLY ALL VALUES BY0.10E+00

SECTION J = 5

I =		1	3	5	7	9	11	13	15	17	19
	D =	1	2000	2000	2000	2000	2000	2000	2000	2000	2000
1	-0.003	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2
2	-0.009	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2
3	-0.018	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2
4	-0.035	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2
5	-0.071	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2
6	-0.125	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2	88.2
7	-0.175	83.9	83.9	83.9	83.9	83.9	83.9	83.9	83.9	83.9	83.9
8	-0.225	79.9	79.9	79.9	79.9	79.9	79.9	79.9	79.9	79.9	79.9
9	-0.275	76.0	76.0	76.0	76.0	76.0	76.0	76.0	76.0	76.0	76.0
10	-0.325	72.3	72.3	72.3	72.3	72.3	72.3	72.3	72.3	72.3	72.3
11	-0.375	68.7	68.7	68.7	68.7	68.7	68.7	68.7	68.7	68.7	68.7
12	-0.425	65.4	65.4	65.4	65.4	65.4	65.4	65.4	65.4	65.4	65.4
13	-0.475	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2
14	-0.525	59.2	59.2	59.2	59.2	59.2	59.2	59.2	59.2	59.2	59.2
15	-0.575	56.3	56.3	56.3	56.3	56.3	56.3	56.3	56.3	56.3	56.3
16	-0.625	53.5	53.5	53.5	53.5	53.5	53.5	53.5	53.5	53.5	53.5
17	-0.675	50.9	50.9	50.9	50.9	50.9	50.9	50.9	50.9	50.9	50.9
18	-0.725	48.4	48.4	48.4	48.4	48.4	48.4	48.4	48.4	48.4	48.4
19	-0.775	46.1	46.1	46.1	46.1	46.1	46.1	46.1	46.1	46.1	46.1
20	-0.825	43.8	43.8	43.8	43.8	43.8	43.8	43.8	43.8	43.8	43.8
21	-0.875	41.7	41.7	41.7	41.7	41.7	41.7	41.7	41.7	41.7	41.7
22	-0.929	39.5	39.5	39.5	39.5	39.5	39.5	39.5	39.5	39.5	39.5
23	-0.975	37.7	37.7	37.7	37.7	37.7	37.7	37.7	37.7	37.7	37.7
24	-1.025	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig. 9 The vertical distribution of initial temperature for the slice at J=5 in the X-Z plane, displayed by CALL PRXZ('TEMP.',TIME,TB,IM,2,JM,5,10,KB,1.E-1,DT,ZZ). Add 10 C° to obtain the real temperature.

RHO

TIME = 0.00 MULTIPLY ALL VALUES BY0.10E-03

SECTION I = 10

J =		1	3	5	7	9	11	13	15	17	19
	D =	1.2000	2000	2000	2000	2000	2000	2000	2000	2000	.
1	-0.003	0.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
2	-0.009	0.0	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
3	-0.018	0.0	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
4	-0.035	0.0	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7
5	-0.071	0.0	6.8	6.8	6.8	6.8	6.8	6.8	6.8	6.8	6.8
6	-0.125	0.0	11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5
7	-0.175	0.0	16.9	16.9	16.9	16.9	16.9	16.9	16.9	16.9	16.9
8	-0.225	0.0	22.2	22.2	22.2	22.2	22.2	22.2	22.2	22.2	22.2
9	-0.275	0.0	27.5	27.5	27.5	27.5	27.5	27.5	27.5	27.5	27.5
10	-0.325	0.0	32.8	32.8	32.8	32.8	32.8	32.8	32.8	32.8	32.8
11	-0.375	0.0	37.9	37.9	37.9	37.9	37.9	37.9	37.9	37.9	37.9
12	-0.425	0.0	43.1	43.1	43.1	43.1	43.1	43.1	43.1	43.1	43.1
13	-0.475	0.0	48.2	48.2	48.2	48.2	48.2	48.2	48.2	48.2	48.2
14	-0.525	0.0	53.2	53.2	53.2	53.2	53.2	53.2	53.2	53.2	53.2
15	-0.575	0.0	58.2	58.2	58.2	58.2	58.2	58.2	58.2	58.2	58.2
16	-0.625	0.0	63.1	63.1	63.1	63.1	63.1	63.1	63.1	63.1	63.1
17	-0.675	0.0	68.1	68.1	68.1	68.1	68.1	68.1	68.1	68.1	68.1
18	-0.725	0.0	72.9	72.9	72.9	72.9	72.9	72.9	72.9	72.9	72.9
19	-0.775	0.0	77.8	77.8	77.8	77.8	77.8	77.8	77.8	77.8	77.8
20	-0.825	0.0	82.6	82.6	82.6	82.6	82.6	82.6	82.6	82.6	82.6
21	-0.875	0.0	87.4	87.4	87.4	87.4	87.4	87.4	87.4	87.4	87.4
22	-0.929	0.0	92.5	92.5	92.5	92.5	92.5	92.5	92.5	92.5	92.5
23	-0.975	0.0	96.8	96.8	96.8	96.8	96.8	96.8	96.8	96.8	96.8
24	-1.025	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig. 10 The vertical distribution of initial density for the slice at I=10 in the Y–Z plane, displayed by
CALL PRYZ('RHO',TIME,RHO,IM,2,JM,2,KB,1.E–4,DT,ZZ).
The actual physical density in $Kg\ m^{-3}$ is found by adding 1.025 to the value and then multiplyng by 1000, eg., $(1.025 + 0.00014) \times 1000 = 1025.14\ Kg\ m^{-3}$.

WUSURF

TIME = 5.0000 DAYS MULTIPLY ALL VALUES BY 1.000E-08

	1	3	5	7	9	11	13	15	17	19
20	0	0	0	0	0	0	0	0	0	0
19	0	-4931	-4931	-4931	-4931	-4931	-4931	-4931	-4931	-4931
18	0	-9458	-9458	-9458	-9458	-9458	-9458	-9458	-9458	-9458
17	0	-8794	-8794	-8794	-8794	-8794	-8794	-8794	-8794	-8794
16	0	-7891	-7891	-7891	-7891	-7891	-7891	-7891	-7891	-7891
15	0	-6772	-6772	-6772	-6772	-6772	-6772	-6772	-6772	-6772
14	0	-5469	-5469	-5469	-5469	-5469	-5469	-5469	-5469	-5469
13	0	-4016	-4016	-4016	-4016	-4016	-4016	-4016	-4016	-4016
12	0	-2454	-2454	-2454	-2454	-2454	-2454	-2454	-2454	-2454
11	0	-825	-825	-825	-825	-825	-825	-825	-825	-825
10	0	825	825	825	825	825	825	825	825	825
9	0	2454	2454	2454	2454	2454	2454	2454	2454	2454
8	0	4016	4016	4016	4016	4016	4016	4016	4016	4016
7	0	5469	5469	5469	5469	5469	5469	5469	5469	5469
6	0	6772	6772	6772	6772	6772	6772	6772	6772	6772
5	0	7891	7891	7891	7891	7891	7891	7891	7891	7891
4	0	8794	8794	8794	8794	8794	8794	8794	8794	8794
3	0	9458	9458	9458	9458	9458	9458	9458	9458	9458
2	0	4931	4931	4931	4931	4931	4931	4931	4931	4931
1	0	0	0	0	0	0	0	0	0	0

Fig. 11 The east–west momentum flux from wind forcing of the box model. It is the negative of the wind stress divided by the water density, expressed in units of $m^2\ s^{-2}$. It is displayed with the command
CALL PRXY('WUSURF',TIME,WUSURF,IM,2,JM,1,1.0E–8).

TIME = 5.0000 DAYS FREE SURVACE
MULTIPLY ALL VALUES BY 1.000E-04

	1	3	5	7	9	11	13	15	17	19
20	0	0	0	0	0	0	0	0	0	0
19	0	-227	-216	-212	-210	-210	-208	-206	-204	-189
18	0	-161	-131	-123	-124	-128	-137	-152	-172	-182
17	0	-96	-44	-33	-35	-45	-61	-86	-121	-162
16	0	-46	25	39	35	21	-2	-37	-88	-153
15	0	-3	86	104	98	80	50	5	-58	-146
14	0	32	137	158	150	128	93	41	-34	-141
13	0	60	176	199	190	166	127	68	-16	-138
12	0	79	203	226	217	190	149	86	-4	-137
11	0	90	216	240	231	203	159	94	1	-137
10	0	91	216	240	230	203	158	94	0	-139
9	0	84	204	227	217	189	147	84	-7	-142
8	0	69	179	200	191	165	125	65	-19	-146
7	0	45	143	162	154	130	93	39	-37	-152
6	0	15	97	114	106	86	53	6	-60	-160
5	0	-21	43	56	50	33	6	-33	-88	-168
4	0	-63	-16	-7	-12	-24	-46	-76	-118	-179
3	0	-123	-94	-88	-91	-100	-114	-136	-164	-199
2	0	-164	-160	-164	-169	-175	-183	-194	-208	-221
1	0	0	0	0	0	0	0	0	0	0

Fig. 12 The box model sea level in meters after 5 days, displayed by CALL PRXY('FREE SURFACE',TIME,ELB,IM,2,JM,1,1.E-4).

AVERAGE U COMP.

TIME = 5.0000 DAYS MULTIPLY ALL VALUES BY 1.000E-04

	1	3	5	7	9	11	13	15	17	19
20	0	0	0	0	0	0	0	0	0	0
19	0	17	74	89	86	76	60	36	3	-28
18	0	106	172	186	184	174	158	135	103	57
17	0	93	159	173	171	161	147	125	94	45
16	0	80	141	155	153	144	131	111	82	36
15	0	67	121	132	131	124	112	95	69	28
14	0	53	97	107	105	100	90	76	55	22
13	0	39	71	78	77	73	66	56	40	15
12	0	23	43	47	47	44	40	34	24	9
11	0	7	14	15	15	15	13	11	8	3
10	0	-8	-15	-16	-16	-15	-13	-11	-8	-3
9	0	-24	-44	-48	-47	-45	-40	-34	-24	-9
8	0	-39	-72	-78	-77	-73	-66	-55	-40	-15
7	0	-54	-98	-107	-106	-100	-90	-76	-55	-22
6	0	-68	-121	-133	-131	-124	-112	-95	-69	-28
5	0	-80	-142	-155	-153	-145	-131	-111	-82	-36
4	0	-93	-159	-173	-171	-162	-147	-125	-93	-45
3	0	-106	-171	-186	-184	-175	-159	-136	-103	-57
2	0	-15	-73	-88	-86	-76	-60	-36	-3	29
1	0	0	0	0	0	0	0	0	0	0

Fig. 13 The box model barotropic east–west current in $m\ s^{-1}$ after 5 days, displayed by CALL PRXY('AVERAGE U COMP.',TIME,UAB,IM,2,JM,1,1.E–4).

TIME = 5.0000 DAYS AVERAGE V COMP.
MULTIPLY ALL VALUES BY 1.000E-04

	1	3	5	7	9	11	13	15	17	19
20	0	0	0	0	0	0	0	0	0	0
19	0	17	5	0	-2	-3	-5	-7	-9	14
18	0	39	10	0	-4	-7	-10	-15	-19	-14
17	0	61	15	0	-6	-10	-15	-22	-30	-36
16	0	81	20	0	-8	-13	-20	-29	-40	-54
15	0	99	24	0	-9	-16	-24	-34	-49	-69
14	0	113	28	0	-10	-18	-27	-39	-56	-80
13	0	124	30	0	-11	-19	-29	-43	-61	-88
12	0	130	32	0	-12	-20	-31	-45	-65	-92
11	0	133	33	0	-12	-21	-31	-45	-66	-94
10	0	130	32	0	-12	-20	-30	-45	-65	-92
9	0	124	31	0	-11	-19	-29	-43	-61	-88
8	0	113	28	0	-10	-18	-27	-39	-56	-80
7	0	99	25	0	-9	-16	-24	-34	-49	-69
6	0	81	20	0	-8	-13	-20	-29	-40	-54
5	0	60	16	0	-6	-10	-15	-22	-30	-36
4	0	39	10	0	-4	-7	-10	-15	-19	-14
3	0	17	5	0	-2	-3	-5	-7	-9	14
2	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0

Fig. 14 The box model barotropic north–south current in $m\ s^{-1}$ after 5 days, displayed by CALL PRXY('AVERAGE V COMP.',TIME,VAB,IM,2,JM,1,1.E–4).

DISTRIBUTION LIST

1. Office of Naval Research
   Code 1242 (10 copies)
   800 North Quincy Street
   Arlington, VA 22217-5000

2. Director, Atmospheric Sciences Directorate
   NOARL West (Code 400)
   Monterey, CA 93943-5000

3. Commanding Officer
   Fleet Numerical Oceanography Office
   Monterey, CA 93943-5000

4. Commanding Officer
   Naval Oceanographic Office
   Stennis Space Center, MS 39529

5. Technical Director
   CNOC (Code OOT)
   Stennis Space Center, MS 39529

6. Commanding Officer
   NOARL (Code 100)
   Stennis Space Center, MS 39529

7. Technical Director
   NOARL (Code 110)
   Stennis Space Center, MS 39529

8. Director, Ocean Sciences Directorate
   NOARL (Code 300)
   Stennis Space Center, MS 39529

9. Head, Ocean Sensing and Prediction Division
   NOARL (Code 320)
   Stennis Space Center, MS 39529

10. Library (3 copies)
    NOARL (Code 125)
    Stennis Space Center, MS 39529

11. Dr. J. Dana Thompson
    NOARL, Code 320
    Building 1103, Room 248
    Stennis Space Center, MS 39529

12. Dr. William J. Schmitz, Jr.
    Department of Oceanography
    Woods Hole Oceanographic Inst.
    Woods Hole, MA 02543

13. Dr. William Holland
    National Center for Atmospheric Research
    P. O. Box 3000
    Boulder, CO 80307

14. UCAR Library
    P. O. Box 3000
    Boulder, CO 80307

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. Agency Use Only *(Leave blank).*	2. Report Date.	3. Report Type and Dates Covered.
	March 1991	Special Project

4. Title and Subtitle.
A User's Manual for the Princeton Numerical Ocean Model

5. Funding Numbers.
Program Element No. 0601153N
Project No. R310300
Task No. 801
Accession No. DN250022

6. Author(s).
William P. O'Connor

7. Performing Organization Name(s) and Address(es).
Institute for Naval Oceanography
Building 1103, Room 233
Stennis Space Center, MS 39529-5005

8. Performing Organization Report Number.
SP-5

9. Sponsoring/Monitoring Agency Name(s) and Address(es).
Naval Oceanographic and Atmospheric Research Laboratory
Code 110, Stennis Space Center, MS 39529

10. Sponsoring/Monitoring Agency Report Number.

11. Supplementary Notes.

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

This is a user's guide for the numerical ocean model developed at Princeton University and presently in use at the Institute for Naval Oceanography. The basic model has been applied successfully to study such diverse regions as Chesapeake Bay (Blumberg 1977), the South Atlantic Bight (Blumberg and Mellor 1983), the Mid Atlantic Bight (Blumberg and Kantha 1985), the Gulf of Mexico (Blumberg and Mellor 1985), New York Harbor (Oey et al 1985a,b,c), Delaware Bay (Galperin and Mellor 1990a,b), and the western Atlantic including the Gulf Stream (Mellor and Ezer 1991). It can model regions from the size of bays and estuaries to basin scale North Atlantic domains. A good discussion of modeling bays and coastal oceans is given by Blumberg and Oey (1985).

14. Subject Terms.	15. Number of Pages.
(U) INO (U) NAOPS (U) PRIDER (U) PEDAM (U) OGCM	69
	16. Price Code.

17. Security Classification of Report.	18. Security Classification of This Page.	19. Security Classification of Abstract.	20. Limitation of Abstract.
Unclassified	Unclassified	Unclassified	SAR

No. 21. Breaker, L. C., 1989: El Nino and Related Variability in Sea-Surface Temperature Along the Central California Coast. PACLIM Monograph of Climate Variability of the Eastern North Pacific and Western North America, Geophysical Monograph 55, AGU, 133-140.

No. 22. Yu, T. W., D. C. Esteva, and R. L. Teboulle, 1991: A Feasibility Study on Operational Use of Geosat Wind and Wave Data at the National Meteorological Center. Technical Note/NMC Office Note No. 380, 28pp.

No. 23. Burroughs, L. D., 1989: Open Ocean Fog and Visibility Forecasting Guidance System. Technical Note/NMC Office Note No. 348, 18pp.

No. 24. Gerald, V. M., 1987: Synoptic Surface Marine Data Monitoring. Technical Note/NMC Office Note No. 335, 10pp.

No. 25. Breaker, L. C., 1989: Estimating and Removing Sensor Induced Correlation form AVHRR Data. Journal of Geophysical Reseach, 95, 9701-9711.

No. 26. Chen, H. S., 1990: Infinite Elements for Water Wave Radiation and Scattering. International Journal for Numerical Methods in Fluids, 11, 555-569.

No. 27. Gemmill, W. H., T. W. Yu, and D. M. Feit, 1988: A Statistical Comparison of Methods for Determining Ocean Surface Winds. Journal of Weather and Forecasting, 3, 153-160.

No. 28. Rao. D. B., 1989: A Review of the Program of the Ocean Products Center. Weather and Forecasting, 4, 427-443.

No. 29. Chen, H. S., 1989: Infinite Elements for Combined Diffration and Refraction . Conference Preprint, Seventh International Conference on Finite Element Methods Flow Problems, Huntsville, Alabama, 6pp.

No. 30. Chao, Y. Y., 1989: An Operational Spectral Wave Forecasting Model for the Gulf of Mexico. Proceedings of 2nd International Workshop on Wave Forecasting and Hindcasting, 240-247.

No. 31. Esteva, D. C., 1989: Improving Global Wave Forecasting Incorporating Altimeter Data. Proceedings of 2nd International Workshop on Wave Hindcasting and Forecasting,Vancouver, B.C., April 25-28, 1989, 378-384.

No. 32. Richardson, W. S., J. M. Nault, and D. M. Feit, 1989: Computer-Worded Marine Forecasts. Preprint, 6th Symp. on Coastal Ocean Management Coastal Zone 89, 4075-4084.

No. 33. Chao, Y. Y., and T. L. Bertucci, 1989: A Columbia River Entrance Wave Forecasting Program Developed at the Ocean Products Center. Techical Note/NMC Office Note 361.

No. 34. Burroughs, L. D., 1989: Forecasting Open Ocean Fog and Visibility. Preprint, 11th Conference on Probability and Statisitcs, Monterey, Ca., 5pp.

No. 35. Rao, D. B., 1990: Local and Regional Scale Wave Models. Proceeding (CMM/WMO) Technical Conference on Waves, WMO, Marine Meteorological of Related Oceanographic Activities Report No. 12, 125-138.

No. 36. Burroughs, L.D., 1991: Forecast Guidance for Santa Ana conditions. Technical Procedures Bulletin No. 391, 11pp.

No. 37. Burroughs, L. D., 1989: Ocean Products Center Products Review Summary. Technical Note/NMC Office Note No. 359. 29pp.

No. 38. Feit, D. M., 1989: Compendium of Marine Meteorological and Oceanographic Products of the Ocean Products Center (revision 1). NOAA Technical Memo NWS/NMC 68.

No. 39. Esteva, D. C., and Y. Y. Chao, 1991: The NOAA Ocean Wave Model Hindcast for LEWEX. Directional Ocean Wave Spectra, Johns Hopkins University Press, 163-166.

No. 40. Sanchez, B. V., D. B. Rao, and S. D. Steenrod, 1987: Tidal Estimation in the Atlantic and Indian Oceans, 3° x 3° Solution. NASA Technical Memorandum 87812, 18pp.

No. 41. Crosby, D. S., L. C. Breaker, and W. H. Gemmill, 1990: A Defintion for Vector Correlation and its Application to Marine Surface Winds. Technical Note/NMC Office Note No. 365, 52pp.

No. 42. Feit, D. M., and W. S. Richardson, 1990: Expert System for Quality Control and Marine Forecasting Guidance. Preprint, 3rd Workshop Operational and Metoerological. CMOS, 6pp.

No. 43. Gerald, V. M., 1990: OPC Unified Marine Database Verification System. Technical Note/NMC Office Note No. 368, 14pp.

No. 44. Wohl, G. M., 1990: Sea Ice Edge Forecast Verification System. National Weather Association Digest, (submitted)

No. 45. Feit, D. M., and J. A. Alpert, 1990: An Operational Marine Fog Prediction Model. NMC Office Note No. 371, 18pp.

No. 46. Yu, T. W., and R. L. Teboulle, 1991: Recent Assimilation and Forecast Experiments at the National Meteorological Center Using SEASAT-A Scatterometer Winds. Technical Note/NMC Office Note No. 383, 45pp.

No. 47. Chao, Y. Y., 1990: On the Specification of Wind Speed Near the Sea Surface. Marine Forecaster Training Manual.

No. 48. Breaker, L. C., L. D. Burroughs, T. B. Stanley, and W. B. Campbell, 1992: Estimating Surface Currents in the Slope Water Region Between 37 and 41°N Using Satellite Feature Tracking. Technical Note, 47pp.

No. 49. Chao, Y. Y., 1990: The Gulf of Mexico Spectral Wave Forecast Model and Products. Technical Procedures Bulletin No. 381, 3pp.

No. 50. Chen, H. S., 1990: Wave Calculation Using WAM Model and NMC Wind. Preprint, 8th ASCE Engineering Mechanical Conference, 1, 368-372.

No. 51. Chao, Y. Y., 1990: On the Transformation of Wave Spectra by Current and Bathymetry. Preprint, 8th ASCE Engineering Mechnical Conference, 1, 333-337.

No. 52. WAS NOT PUBLISHED

No. 53. Rao, D. B., 1991: Dynamical and Statistical Prediction of Marine Guidance Products. Proceedings, IEEE Conference Oceans 91, 3, 1177-1180.

No. 54. Gemmill, W. H., 1991: High-Resolution Regional Ocean Surface Wind Fields. Proceedings, AMS 9th Conference on Numerical Weather Prediction, Denver, CO, Oct. 14-18, 1991, 190-191.

No. 55. Yu, T. W., and D. Deaven, 1991: Use of SSM/I Wind Speed Data in NMC's GDAS. Proceedings, AMS 9th Conference on Numerical Weather Prediction, Denver, CO, Oct. 14-18, 1991, 416-417.

No. 56. Burroughs, L. D., and J. A. Alpert, 1993: Numerical Fog and Visiability Guidance in Coastal Regions. Technical Procedures Bulletin. No. 398, 6pp.

No. 57. Chen, H. S., 1992: Taylor-Gelerkin Method for Wind Wave Propagation. ASCE 9th Conf. Eng. Mech. (in press)

No. 58. Breaker, L. C., and W. H. Gemmill, and D. S. Crosby, 1992: A Technique for Vector Correlation and its Application to Marine Surface Winds. AMS 12th Conference on Probability and Statistics in the Atmospheric Sciences, Toronto, Ontario, Canada, June 22-26, 1992.

No. 59. Yan, X.-H., and L. C. Breaker, 1993: Surface Circulation Estimation Using Image Processing and Computer Vision Methods Applied to Sequential Satellite Imagery. Photogrammetric Engineering and Remote Sensing, 59, 407-413.

No. 60. Wohl, G., 1992: Operational Demonstration of ERS-1 SAR Imagery at the Joint Ice Center. Proceeding of the MTS 92 - Global Ocean Partnership, Washington, DC, Oct. 19-21, 1992.

No. 61. Waters, M. P., Caruso, W. H. Gemmill, W. S. Richardson, and W. G. Pichel, 1992: An Interactive Information and Processing System for the Real-Time Quality Control of Marine Meteorological Oceanographic Data. Pre-print 9th International Conference on Interactive Information and Processing System for Meteorology, Oceanography and Hydrology, Anaheim, CA, Jan. 17-22, 1993.

No. 62. Breaker, L. C., and V. Krasnopolsky, 1994: The Problem of AVHRR Image Navigation Revisited. Int. Journal of Remote Sensing, 15, 979-1008.

No. 63. Crosby, D. S., L. C. Breaker, and W. H. Gemmill, 1993: A Proposed Definition for Vector Correlation in Geophysics: Theory and Application. Journal of Atmospheric and Ocean Technology, 10, 355-367.

No. 64. Grumbine, R., 1993: The Thermodynamic Predictability of Sea Ice. Journal of Glaciology, 40, 277-282, 1994.

No. 65. Chen, H. S., 1993: Global Wave Prediction Using the WAM Model and NMC Winds. 1993 International Conference on Hydro Science and Engineering, Washington, DC, June 7 - 11, 1993. (submitted)

No. 66. WAS NOT PUBLISHED

No. 67. Breaker, L. C., and A. Bratkovich, 1993: Coastal-Ocean Processes and their Influence on the Oil Spilled off San Francisco by the M/V Puerto Rican. Marine Environmental Research, 36, 153-184.

No. 68. Breaker, L. C., L. D. Burroughs, J. F. Culp, N. L. Gunasso, R. Teboulle, and C. R. Wong, 1993: Surface and Near-Surface Marine Observations During Hurricane Andrew. Technical Note/NMC Office Note #398, 41pp.

No. 69. Burroughs, L. D., and R. Nichols, 1993: The National Marine Verification Program - Concepts and Data Management, Technical Note/NMC Office Note #393, 21pp.

No. 70. Gemmill, W. H., and R. Teboulle, 1993: The Operational Use of SSM/I Wind Speed Data over Oceans. Pre-print 13th Conference on Weather Analyses and Forecasting, AMS Vienna, VA., August 2-6, 1993, 237-238.

No. 71. Yu, T.-W., J. C. Derber, and R. N. Hoffman, 1993: Use of ERS-1 Scatterometer Backscattered Measurements in Atmospheric Analyses. Pre-print 13th Conference on Weather Analyses and Forecasting, AMS, Vienna, VA., August 2-6, 1993, 294-297.

No. 72. Chalikov, D. and Y. Liberman, 1993: Director Modeling of Nonlinear Waves Dynamics. J. Physical, (To be submitted).

No. 73. Woiceshyn, P., T. W. Yu, W. H. Gemmill, 1993: Use of ERS-1 Scatterometer Data to Derive Ocean Surface Winds at NMC. Pre-print 13th Conference on Weather Analyses and Forecasting, AMS, Vienna, VA, August 2-6, 1993, 239-240.

No. 74. Grumbine, R. W., 1993: Sea Ice Prediction Physics. Technical Note/NMC Office Note #396, 44pp.

No. 75. Chalikov, D., 1993: The Parameterization of the Wave Boundary Layer. Journal of Physical Oceanography, Vol. 25, No. 6, Par 1, 1333-1349.

No. 76. Tolman, H. L., 1993: Modeling Bottom Friction in Wind-Wave Models. Ocean Wave Measurement and Analysis, O.T. Magoon and J.M. Hemsley Eds., ASCE, 769-783.

No. 77. Breaker, L., and W. Broenkow, 1994: The Circulation of Monterey Bay and Related Processes. Oceanography and Marine Biology: An Annual Review, 32, 1-64.

No. 78. Chalikov, D., D. Esteva, M. Iredell and P. Long, 1993: Dynamic Coupling between the NMC Global Atmosphere and Spectral Wave Models. Technical Note/NMC Office Note #395, 62pp.

No. 79. Burroughs, L. D., 1993: National Marine Verification Program - Verification Statistics - Verification Statistics, Technical Note/NMC Office Note #400, 49 pp.

No. 80. Shashy, A. R., H. G. McRandal, J. Kinnard, and W. S. Richardson, 1993: Marine Forecast Guidance from an Interactive Processing System. 74th AMS Annual Meeting, January 23 - 28, 1994.

No. 81. Chao, Y. Y., 1993: The Time Dependent Ray Method for Calculation of Wave Transformation on Water of Varying Depth and Current. Wave 93 ASCE.

No. 82. Tolman, H. L., 1994: Wind-Waves and Moveable-Bed Bottom Friction. Journal of Physical Oceanography, 24, 994-1009.

No. 83. Grumbine, R. W., 1993: Notes and Correspondence A Sea Ice Albedo Experiment with the NMC Meduim Range Forecast Model. Weather and Forecasting, (submitted).

No. 84. Chao, Y. Y., 1993: The Gulf of Alaska Regional Wave Model. Technical Procedure Bulletin, No. 427, 10 pp.

No. 85. Chao, Y. Y., 1993: Implementation and Evaluation of the Gulf of Alaska Regional Wave Model. Technical Note, 35 pp.

No. 86. WAS NOT PUBLISHED.

No. 87. Burroughs, L., 1994: Portfolio of Operational and Development Marine Meteorological and Oceanographic Products. Technical Note/NCEP Office Note No. 412, 52 pp. [PB96-158548]

No. 88. Tolman, H. L., and D. Chalikov, 1994: Development of a third-generation ocena wave model at NOAA-NMC. Proc. Waves Physical and Numberial Modelling, M. Isaacson and M.C. Quick Eds., Vancover, 724-733.

No. 89. Peters, C., W. H. Gemmill, V. M. Gerald, and P. Woiceshyn, 1994: Evaluation of Empirical Transfer Functions for ERS-1 Scatterometer Data at NMC. 7th Conference on Satellite Meteorology and Oceanography, June 6-10, 1994, Monterey, CA., pg. 550-552.

No. 90. Breaker, L. C., and C. R. N. Rao, 1996: The Effects of Aerosols from the Mt. Pinatubo and Mt. Hudson Volcanio Eruption on Satellite-Derived Sea Surface Temperatures. Journal of Geophysical Research. (To be submitted).

No. 91. Yu, T-W., P. Woiceshyn, W. Gemmill, and C. Peters, 1994: Analysis & Forecast Experiments at NMC Using ERS-1 Scatterometer Wind Measurements. 7th Conference on Satellite Meteorology and Oceanography, June 6-10, 1994, Monterey, CA., pg. 600-601.

No. 92. Chen, H. S., 1994: Ocean Surface Waves. Technical Procedures Bulletin, No. 426, 17 pp.

No. 93. Breaker, L. C., V. Krasnopolsky, D. B. Rao, and X.-H. Yan, 1994: The Feasibility of Estimating Ocean Surface Currents on an Operational Basis using Satellite Feature Tracking Methods. Bulletin of the American Meteorolgical Society, 75, 2085-2095.

No. 94. Krasnopolsky V., L. C. Breaker, and W. H. Gemmill, 1994: Development of Single "All-Weather" Neural Network Algorithms for Estimating Ocean Surface Winds from the Special Sensor Microwave Imager. Technical Note.

No. 95. Breaker, L. C., D. S. Crosby and W. H. Gemmill, 1994: The application of a New Definition for Vector Correlation to Problems in Oceanography and Meteorology. Journal of Applied Meteolorology, 33, 1354-1365.

No. 96. Peters, C. A., V. M. Gerald, P. M. Woiceshyn, and W. H. Gemmill, 1994: Operational Processing of ERS-1 Scatterometer winds: A Documentation. Technical Note.

No. 97. Gemmill, W. H., P. M. Woiceshyn, C. A. Peters, and V. M. Gerald, 1994: A Preliminary Evaluation Scatterometer Wind Transfer Functions for ERS-1 Data. Technical Note.

No. 98. Chen, H. S., 1994: Evaluation of a Global Ocean Wave Model at NMC. International Conference on Hydro-Science and Engineering. Beijing, China, March 22 - 26, 1995.

No. 99. Aikman, F. and D. B. Rao, 1994: NOAA Perspective on a Coastal Forecast System.

No. 100. Rao, D. B. and C. Peters, 1994: Two-Dimensional Co-Oscillations in a Rectangular Bay: Possible Application to Water Problems. OPC Office Note.

No. 101. Breaker, L. C., L. D. Burroughs, Y. Y. Chao, J. F. Culp, N. L. Gunasso, R. Teboulle, and C. R. Wong, 1994: Surface and Near-Surface Marine Observations During Hurricane Andrew. Weather and Forecasting, 9, 542-556.

No. 102. Tolman, H. L., 1995: Subgrid Modeling of Moveable-bed Bottom Friction in Wind Wave Models. Coastal Engineering, (in press).

No. 103. Breaker, L. C., D. B. Gilhousen, H. L. Tolman and L. D. Burroughs, 1995: Initial Results from Long-Term Measurements of Atmospheric Humidity and Related Parameters the Marine Boundary Layer at Two Locations in the Gulf of Mexico. (To be submitted to Global Atmosphere and Ocean Systems).

No. 104. Burroughs, L. D., and J. P. Dallavalle, 1995: Great Lakes Wind and Wave Guidance. Technical Procedures Bulletin No., (In preparation).

No. 105. Burroughs, L. D., and J. P. Dallavalle, 1995: Great Lakes Storm Surge Guidance. Technical Procedures Bulletin No., (In preparation).

No. 106. Shaffer, W. A., J. P. Dallavalle, and L. D. Burroughs, 1995: East Coast Extratropical Storm Surge and Beach Erosion Guidance. Technical Procedures Bulletin No., (In preparation)

No. 107. WAS NOT PUBLISHED.

No. 108. WAS NOT PUBLISHED.

No. 109. WAS NOT PUBLISHED.

No. 110. Gemmill, W. H, and C. A. Peters, 1995: The Use of Satellite Dervired Wind Data in High-Resolution Regional Ocean Surface Wind Fields. Conference on Coastal Oceanic and Atmospheric Prediction, Jan 28 - Feb 2, 1996, Atlanta, GA (accepted at preprint press).

OPC CHANGES TO OMB

No. 111. Krasnopolsky, V. M, W. H. Gemmill, and L. C. Breaker, 1995: Improved SSM/I Wind Speed Retrievals at Higher Wind Speeds. Journal of Geophysical Research, (in press).

No. 112. Chalikov, D., L. D. Breaker, and L. Lobocki, 1995: A Simple Model of Mixing in the Upper Ocean. Journal of Physical Ocean, (in press).

No. 113. Tolman, H. L., 1995: On the Selection of Propagation Schemes for a Spectral Wind-Wave Model. NCEP Office Note No. 411.

No. 114. Grumbine, R. W., 1995: Virtual Floe Ice Drift Forecast Model Intercomparison. NCEP Office Note. (To be submitted).

No. 115. Grumbine, R. W., 1995: Sea Ice Forecast Model Intercomparison: Selecting a Base Model for NCEP Sea Ice Modelling. Technical Note.

No. 116. Yu, T. W. and J. C. Derber, 1995: Assimilation Experiments with ERS-1 Winds: Part I - Use of Backscatter Measurements in the NMC Spectral Statistical Analysis System. Technical Note.

No. 117. Yu, T. W., 1995: Assimilation Experiments with ERS1 Winds: Part II - Use of Vector Winds in NCEP Spectral Statistical Analysis System. Technical Note.

No. 118. Grumbine, R. W., 1995: Sea Ice Drift Guidance. Technical Procedures Bulletin. (submitted)

No. 119. Tolman, H. L., 1996: Statistical Model Validation Techniques Applied to Marine Wind Speed Analysis. Technical Note.

No. 120. Grumbine, R. W., 1996: Automated Passive Microwave Sea Ice Concentration Analysis at NCEP. Technical Note.

No. 121. Grumbine, R. W., 1996: Sea Ice Prediction Environment: Documentation. Technical Note.

No. 122. Tolman, H. L and D. Chalikov, 1996: On the Source Terms in a Third-Generation Wind Wave Model. Journal of Physical Oceanography. (To be submitted).

No. 123. Gemmill, W. H., V. Krasnopolsky, L. C. Breaker, and C. Peters, 1996: Developments to Improve Satellite Derived Ocean Surface Winds for use in Marine Analyses. Pre-print Numerical Weather Prediction Conference, Norfolk, VA, Aug. 19-23, 1996 (To be submitted).

No. 124. Breaker, L. C., D. B. Gilhousen, H. L. Tolman and L. D. Burroughs, 1996: Initial Results from Long-Term Measurements of Atmospheric Humidity and Related Parameters in the Marine Boundary Layer at Two Locations in the Gulf of Mexico. NCEP Office Note No. 414.

No. 125. Yu, T. W., M. D. Iredell, and Y. Zhu, 1996: The Impact of ERS-1 Winds on NCEP Operational Numerical Weather Analyses and Forecast. Pre-print Numerical Weather Predicition Conference, Norfolk, VA, August 19-23, 1996. (To be submitted).

No. 126. Burroughs, L. D., 1996: Marine Meteorological and Oceanographic Guidance Products from the National Centers for Environmental Prediction. Mariners Weather Log. (To be submitted).


DEPARTMENT OF COMMERCE
UNITED STATES OF AMERICA