Software Design Document for the Polar Ice Prediction System Version 2.0

RUTH H. PRELLER
PAMELA G. POSEY

Ocean Dynamics and Prediction Branch
Oceanography Division

December 15, 1997

19980127 032

Approved for public release; distribution unlimited.
Since 1987, the Fleet Numerical Meteorology and Oceanography Center (FNMOC) has been running sea ice forecasting systems in various regions of Navy interest (the Central Arctic, the Barents Sea, and the Greenland Sea). The Polar Ice Prediction System (PIPS1.1) predicts sea ice conditions in the Arctic basin, the Barents Sea, and the Greenland Sea at a resolution of 127 km. Two regional sea ice forecasting systems, the Polar Ice Prediction System – Barents Sea (RPIPS-B) and the Polar Ice Prediction System – Greenland Sea (RPIPS-G), also predict sea ice conditions in the Barents and Greenland Seas, respectively, at a higher resolution of 20–25 km. In 1995, the Naval Research Laboratory delivered to FNMOC a coupled ice-ocean system, the Polar Ice Prediction System 2.0 (PIPS2.0), which predicts sea ice conditions of most of the ice-covered regions in the Northern Hemisphere. PIPS2.0 will replace the three existing operational forecast systems when it completes the final operational testing phase at FNMOC. PIPS2.0 uses as its basis the Hibler ice model and the Cox ocean model. PIPS2.0 has a resolution of approximately a quarter of a degree, which is similar to the resolution of the operational regional systems (RPIPS-B and RPIPS-G). This report briefly describes the main program and each of its subroutines.
CONTENTS

1.0 INTRODUCTION .............................................................................................................. 1

   1.1 Scope ......................................................................................................................... 1
   1.2 System Overview ..................................................................................................... 1
   1.3 Overview .................................................................................................................. 2

2.0 GENERAL DESIGN ....................................................................................................... 2

   2.1 CSCI Overview ....................................................................................................... 2
   2.2 CSCI Design Description ....................................................................................... 6

3.0 DETAILED DESIGN ................................................................................................... 12

   3.1 CSC Program Driver.............................................................................................. 12
   3.2 CSC Input ............................................................................................................... 15
   3.3 CSC Compute Ice ................................................................................................. 20
   3.4 CSC Compute Ocean ........................................................................................... 69
   3.5 CSC Next Timestep .............................................................................................. 110
   3.6 CSC Output ........................................................................................................... 112

4.0 PIPS2.0 DATA TABLES .......................................................................................... 114

   4.1 Ice Data Table ....................................................................................................... 115
   4.2 Ocean Data Table ................................................................................................. 122

5.0 CSCI PIPS2.0 DATA FILES ................................................................................... 145

   5.1 Data File to CSU/CSC Cross-Reference ................................................................ 145
   5.2 CSU ODam Virtual Disk Access Logical Units ....................................................... 146

6.0 ABBREVIATIONS AND ACRONYMS ...................................................................... 147

7.0 SUMMARY AND CONCLUSION .............................................................................. 148

8.0 ACKNOWLEDGMENTS ............................................................................................ 148

9.0 REFERENCES ............................................................................................................ 148
SOFTWARE DESIGN DOCUMENT FOR
THE POLAR ICE PREDICTION SYSTEM VERSION 2.0

1.0 INTRODUCTION

1.1 Scope

This Software Design Document (SDD) describes the Computer Software Configuration Item (CSCI) identified as the Polar Ice Prediction System 2.0 (PIPS2.0). This SDD has been prepared in accordance with guidelines set forth by the Fleet Numerical Meteorology and Oceanography Center (FNMOC). These guidelines are based on the Data Item Description (DID) DI-MCCR-80012A of DOD-STD-2167A.

1.2 System Overview

PIPS2.0 was developed as an ice-ocean coupled model to provide daily forecasts of ice-drift velocity, ice thickness, and ice concentration for most ice-covered regions in the Northern Hemisphere. (This includes the area from the North Pole south to approximately 30° N.)

Two independent models were merged to form PIPS2.0: the Hibler Viscous-Plastic Sea Ice Model, which provides the ice prediction output, and the Cox Ocean Model, which provides the ocean forcing required as input for the Hibler ice model. To accomplish the merger, the models were first independently adapted to the required prediction basin and then joined by a common driver routine. Information between the coupled models is exchanged via common blocks.

In this configuration, the ocean model provides daily predictions of mixed-layer temperatures, variable freezing temperatures, oceanic heat fluxes, and ocean currents to the ice model, while the ice model supplies the ocean model with ice concentration, ice growth rate, ice thickness, ice thickness growth rate, ice surface temperature, ice-drift velocity, and heat above the freezing temperature.

The basis of the ice model is the Hibler dynamic/thermodynamic ice model (Hibler 1979; 1980) that was modified for operation in the polar regions (Preller and Posey 1989) and updated for spherical coordinates (Cheng and Preller 1992). The ocean model is the Cox primitive equation, numerical model (Cox 1984) that predicts horizontal and vertical velocities, temperature, and salinity for a three-dimensional ocean basin. Daily Naval Operational Global Atmospheric Prediction System (NOGAPS) data supplies the atmospheric data for forcing.

The PIPS2.0 ice-ocean coupled system is presently in research and development at the Naval Research Laboratory (NRL), Stennis Space Center, MS, and is being implemented at FNMOC. The model is designed to run on a UNIX host platform. For greater applicability, host system specifics are kept to a minimum in this document.
1.3 Overview

This SDD provides a complete description of the design of CSCI PIPS2.0. It describes PIPS2.0 as composed of Computer Software Components (CSC) and Computer Software Units (CSU) and details the allocation of requirements from PIPS2.0 to its CSCs and CSUs. The overview of the design requirements of each PIPS2.0 CSC is contained in Sec. 2.0. Section 3.0 contains the detailed design information for each CSC and CSU. Global data elements are listed and described in Sec. 4.0, and Sec. 5.0 provides a data file to CSU cross reference. Abbreviations and acronyms used in this document are listed in Sec. 6.0.

2.0 GENERAL DESIGN

2.1 CSCI Overview

The PIPS2.0, is based on the Hibler sea ice model coupled with the Cox ocean model. PIPS2.0 takes the place of all operational versions of the ice model at FNMOC.

The Hibler ice model (1979; 1980) was originally developed as a stand-alone program to predict ice-drift velocity, ice thickness, and ice concentration on a daily basis. The Cox ocean model (1984) is coupled with the ice model to provide the ocean currents, temperature, and salinity values. The ocean model provides oceanic forcing, including ocean currents, temperature, salinity, and oceanic heat fluxes to the ice model. The ice model provides the ocean model with predictions of ice conditions including ice concentration, ice growth rate, ice thickness, ice thickness growth rate, ice surface temperature, ice-drift velocity, and heat above the freezing temperature. The two models exchange information via common blocks. Ice and ocean restart data are updated on a daily basis—output one day, input the next.

All the external interfaces to PIPS2.0 are shown in Fig. 2.1-1. The first three interfaces in the second row of input are all output from the previous day’s run. Run-specific parameters are user input. Additional model disk file interfaces include subsets of the Levitus climatology data base, river discharge rates, and the NOGAPS data. Levitus climatology and river discharge rates vary monthly, while NOGAPS data vary daily. Land/sea masks and Earth-oriented latitudes and longitudes that correspond to the ocean basin grid cells (Sec. 2.2) are also input.

The following list is a summary of the external interfaces:

Ice Restart File

- **Previous Run** – The ice conditions including ice drift, ice thickness, ice (or water) temperature, and ice coverage are read in as restart data.

- **Current Run** – The ice conditions including ice drift, ice thickness, ice (or water) temperature, and ice coverage are output. This file becomes the restart file for the next PIPS2.0 run.

Ocean Restart File

- **Previous Run** – The ocean conditions including currents, temperature, and salinity for the last two timesteps computed during the previous model run are read in as restart data.

- **Current Run** – The ocean conditions of currents, temperature, and salinity computed for the last two timesteps during the current run are output. This file becomes the restart file for the next PIPS2.0 run.
Ocean Data – Ocean currents and ice growth rate data (including current, temperature, salinity, and ice growth rate) are used to initialize the time-stepping.

Data File for Plotting – Fields of wind, ice thickness, ice drift, and ice concentration are output to an ASCII file for subsequent plotting.

Land/Sea Masking – The land/sea masks for thermodynamic fields, velocity files, and outflow grid cells are defined.

Latitude/Longitude in Earth-Oriented Spherical Coordinates – Grid positions provide Earth-oriented latitudes of U,V points needed for computing Coriolis force.

Monthly Levitus Temperature – Levitus ocean temperature data are interpolated to PIPS2.0 grid.

Monthly Levitus Salinity – Levitus ocean salinity data are interpolated to PIPS2.0 grid.

Monthly River Temperature Data – Monthly river discharge rates and temperatures are provided for 88 grid locations.

NOGAPS Atmospheric Forcing – NOGAPS daily atmospheric data include air temperature, surface forcing, surface vapor pressure, incoming solar radiation, total heat flux, and sensible heat flux to calculate geostrophic winds for forcing.

Run Specific Parameters – The number of timesteps for run, the interval at which to plot, the interval at which to print, the restart indicator, and the run date-time group are user input.

2.1.1 CSCI Architecture

The top-level architecture of PIPS2.0 is illustrated in Fig. 2.1-2. Each block of the figure represents a CSC described in Sec. 2.2.
Fig. 2.1-2 — CSCI top-level architecture
Table 2.1-1 — CSCI Memory Requirements

<table>
<thead>
<tr>
<th>Common Block</th>
<th>Number of Bytes</th>
<th>Common Block</th>
<th>Number of Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BG</td>
<td>67,910,480</td>
<td>PRESS</td>
<td>518,400</td>
</tr>
<tr>
<td>CORSP</td>
<td>521,284</td>
<td>OCEANS</td>
<td>518,400</td>
</tr>
<tr>
<td>COX2</td>
<td>4,152,960</td>
<td>ONEDIM</td>
<td>568,156</td>
</tr>
<tr>
<td>CURNTS</td>
<td>1,555,200</td>
<td>RAD</td>
<td>521,284</td>
</tr>
<tr>
<td>DIFFU3</td>
<td>1,440</td>
<td>RFOR</td>
<td>2,606,420</td>
</tr>
<tr>
<td>FIELDS</td>
<td>2,073,600</td>
<td>RFOR2</td>
<td>2,073,616</td>
</tr>
<tr>
<td>FORCE</td>
<td>1,031,048</td>
<td>RSTRT</td>
<td>7,752,992</td>
</tr>
<tr>
<td>FULLWD</td>
<td>34,996</td>
<td>RVR</td>
<td>1,412</td>
</tr>
<tr>
<td>GROW</td>
<td>518,400</td>
<td>STEPSP</td>
<td>2,884</td>
</tr>
<tr>
<td>LEVITUS</td>
<td>15,638,520</td>
<td>TSTOP</td>
<td>16,594,568</td>
</tr>
<tr>
<td>MASK</td>
<td>1,552,324</td>
<td>WORKSP (R)</td>
<td>4,148,640</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>130,298,436</td>
</tr>
</tbody>
</table>

2.1.2 System States and Modes

The ice model CSCI operates in only one state and in one mode: restart. In the restart mode, the inputs of ice drift, thickness, compactness, lateral heat, and ice temperature are read from the most recent model forecast, or from climatology, if the forecast restart is not available.

Provisions are made to allow for the future vectorization of parts of several CSCs including CSC Compute Internal Mode and Vorticity Driving Function, CSC Compute Tracers, and CSC Compute External Mode.

2.1.3 Memory and Processing Time

Most of the CSCI internal data is located in common blocks with only a relatively small amount of data designated as local to any particular CSC. For this reason, the memory requirements of each CSC are not addressed individually. The CSCI PIPS2.0 memory requirements are given in Table 2.1-1. The amount of local data contained in any of the CSCs is negligible in comparison to the total data stored in common areas. As an estimate, it is sufficient to say that the largest amount of memory capacity required for any CSC is about 2 Mb greater than the 130.3-Mb total given at the bottom of the table.

Because the external modes are computed after the row-by-row computations are performed by CSC Compute Internal Mode and Vorticity Driving Function, the common block WORKSP is redefined in CSC Compute External Mode. Therefore, the memory requirement given for common block WORKSP is the maximum of its two versions, which in this case is the WORKSP defined in CSC Compute External Mode. The common block WORKSP defined in CSC Compute External Mode is referred to as WORKSP (R) to distinguish it from the WORKSP common block used by the other CSCs. CSCs containing fewer than 100 bytes are not shown.
It is important to note that the memory requirements given here are for PIPS2.0 as it is currently configured as a coupling of the ocean and ice models. These memory requirements change significantly based on two factors:

1. If the program is not to be run in a core-contained mode, common block BG can be taken out of CSC Direct Access Manager, reducing the memory requirements by approximately 68 Mb. When the program is not run in a core-contained mode, data that would normally be resident in common block BG are output to disk files. The cost, however, is a substantial increase in execution time.

2. The majority of arrays are dimensioned according to parameters that define the number of grid cells contained in the defined model basin. Table 2-1.1 shows the amount of memory required to process the defined model basin. This model basin is comprised of 360 grid cells meridionally by 360 grid cells zonally by 15 grid cells vertically (or 1,944,000 grid cells). Because a multitude of data elements are associated with each grid cell, the amount of required memory varies significantly with the number of grid cells in the model basin.

2.2 CSC Design Description

PIPS2.0 provides predictions for a model basin that extends from 30° N to the North Pole. The basin is divided into \( 359 \times 359 \) grid cells based on a spherical coordinate system. Each grid cell has an arc length of 0.57°. To remove computational instabilities, the Earth-oriented coordinates are transformed into a new geometry. This is accomplished by first rotating the Earth’s longitudes so that the Prime Meridian is located at 190° E. Then the North Pole is rotated 90° down the 100° E meridian until it resides on the true equator. In this transformed system, the coordinates have the 170° W to 10° E great circle, which passes through the North Pole as the "new" (repositioned) equator.

2.2.1 CSC Program Driver

This CSC serves as the main driving routine for PIPS2.0. It must establish the proper data in memory needed to restart the model each day, drive the ice and ocean CSCs, and save the data needed for the next model restart.

2.2.2 CSC Input

CSC Input is used for reading and/or calculating inputs necessary for the operation of PIPS2.0. It reads in and converts the date-time group to its components, reads in boundary mask data, and reads in the ice and ocean model restart data files from the previous PIPS2.0 run.

2.2.3 CSC Compute Ice

CSC Compute Ice must provide all daily ice data predictions required by FNMO and must provide data required by CSC Compute Ocean. CSC Compute Ice must contain the capability to compute the following elements:

- ice-drift velocity
- ice thickness
- ice thickness growth rate
- ice concentration
- heat above the freezing temperature
- oceanic heat fluxes
- geostrophic winds
2.2.3.1 CSC Atmospheric Forcing

CSC Atmospheric Forcing reads the atmospheric forcing fields that are derived from the 1992 results of NOGAPS. Geostrophic winds are calculated from the input fields.

2.2.3.2 CSC Momentum Balance

The Momentum Balance CSC calculates the terms needed to solve the momentum equation. It estimates the x and y components of forcing due to the ocean currents plus the ice pressure gradient, which are then used to solve the momentum equation through the relaxation method.

2.2.3.3 CSC Ice Rheology

The Ice Rheology CSC calculates the ice stress that is directly related to the ice strength and strain rates. It calculates strain rates, divergence, and viscosities based on plastic flow specified by an elliptic yield curve.

2.2.3.4 CSC Ice Thickness Distribution

The Ice Thickness Distribution CSC accounts for the changes in ice thickness and concentration due to growth, advection, and deformation of the ice. Only thick and thin ice are considered for these calculations. The deformation of thick ice can create thin ice by divergence, while the thin ice can be removed by convergence. Both growth and melt affect the amount of thick and thin ice. While growth can significantly decrease the amount of thin ice and increase the amount of thick ice, melting can add to the amount of thin ice or create open water by decreasing the thick ice. The advection of the ice thickness and compactness due to explicit timestepping is performed. The diffusion of the ice thickness, compactness, and concentration is determined by using explicit forward time differencing. The negative ice to be melted is calculated, and the changes of thickness and compactness for each timestep are then estimated.

2.2.3.5 CSC Ice Strength

The ice strength, as a function of ice thickness distribution, is calculated in CSC Ice Strength. The strength of the ice depends on the amount of thin ice and is calculated in the FORM CSU.

2.2.3.6 CSC Heat Balance

The Heat Balance CSC takes the FNMOC forcing and calculates the terms needed for the heat budget and uses these terms to compute the growth rates of thick and thin ice. It also computes the surface temperature of ice by iteration. This temperature balances the surface heat budget and dictates the conduction of heat through the ice and, therefore, the growth rates.

2.2.4 CSC Compute Ocean

CSC Compute Ocean must provide the oceanic forcing required for CSC Compute Ice. This includes daily predictions of mixed-layer temperatures, variable freezing temperatures, oceanic heat fluxes, and ocean currents. The CSC Compute Ocean must be configurable so that predictions can be provided for a model basin that is compatible with any basin processed by CSC Compute Ice.
2.2.4.1 CSC Ocean Timestep

CSC Ocean Timestep must initialize all quantities needed to begin processing for each timestep and must control the row-by-row computation of prognostic variables for each timestep. Data for each timestep is processed in rows, with a row being a three-dimensional vertical slice or "slab" of the ocean basin comprised of i, j, k grid cells of constant index j with index i varying from western to eastern basin boundaries and with index k varying from the ocean surface to the bottom. (See Fig. 2.2-1 for the layout of the horizontal and vertical grid spacing and indexing.)

Quantities initialized by CSC Ocean Timestep before row-by-row computations can begin are those needed for analysis of the calculation including, for example, change of variance of tracers, rates of change of kinetic energy, and mass transport of tracers. For each row processed, CSC Ocean Timestep establishes the proper data in memory before passing control to CSC Compute Internal Mode and Vorticity Driving Function and CSC Compute Tracers.

2.2.4.1.1 CSC Compute Wind/Ice Stress – CSC Compute Wind/Ice Stress must compute daily wind stress at the sea ice surface and the stress between the sea ice and the mixed layer. This computation is required at the beginning of each timestep processed.

2.2.4.1.2 CSC Load Normalization Constants – CSC Load Normalization Constants loads constants and coefficients for use in CSC Compute Densities. Arrays must be filled with coefficients of the equation of state, and normalizing temperatures and salinities.

2.2.4.1.3 CSC Compute Densities – CSC Compute Densities must provide the normalized densities for all grid cells in the bootstrap row processed by CSC Ocean Timestep and for all rows processed by CSC Compute Tracers and CSC Compute Internal Mode and Vorticity Driving Function. The grid cell densities are required for purposes of checking vertical stability.

2.2.4.1.4 CSC Print Matrix – CSC Print Matrix will print two-dimensional arrays as required by CSC Ocean Timestep. It must accept variable size arrays and must print contiguous array elements as specified by CSC Ocean Timestep.

2.2.4.2 CSC Compute Internal Mode and Vorticity Driving Function

For each row processed by CSC Ocean Timestep, CSC Compute Internal Mode and Vorticity Driving Function computes the internal mode components of the U and V velocities and the vorticity driving function for use in computing the external mode velocity. The terms U and V represent the zonal and meridional components, respectively, of velocity. In the horizontal plane, U and V are situated at the grid cell corners. In the vertical plane, U and V are located halfway through the vertical dimension of the grid cell.

CSC Compute Internal Mode and Vorticity Driving Function must compute the factors that contribute to the internal mode component of velocity including total advection of momentum, horizontal and vertical diffusion of momentum, Coriolis force, and hydrostatic pressure components. The following equations, given in simplified terms, are used to calculate velocity internal mode components:

\[ UA_{\tau} = UA_{\tau-1} + 2 \star DTUV \star (UAM + UHD + UVD + UCF + UHP) \]  \hspace{1cm} (1)

\[ VA_{\tau} = VA_{\tau-1} + 2 \star DTUV \star (VAM + VHD + VVD + VCF + VHP) \]  \hspace{1cm} (2)
Fig. 2.2.1 — (a) Horizontal and (b) vertical grid spacing and indexing
where

\( UA_{\tau u}, UB_{\tau u}, VA_{\tau u}, \) and \( VA_{\tau u-1} \) represent zonal and meridional internal mode components of horizontal velocity for present and previous timesteps,

\( DTUV \) is the length in seconds of the timestep on \( U, V, \)

\( UAM \) and \( VAM \) are zonal and meridional components of total advection of momentum,

\( UHD \) and \( VHD \) are zonal and meridional components of horizontal diffusion of momentum,

\( UVD \) and \( VVD \) are zonal and meridional components of vertical diffusion of momentum,

\( UCF \) and \( VCF \) are zonal and meridional components of Coriolis force, and

\( UHP \) and \( VHP \) are zonal and meridional components of hydrostatic pressure.

### 2.2.4.3 CSC Compute Tracers

CSC Compute Tracers will compute ocean model tracer elements, including temperature and salinity, for each row processed by CSC Ocean Timestep. Tracer elements represented by \( T \) are located at the horizontal center of the ocean model grid cells and halfway through the vertical dimension.

CSC Compute Tracers must compute components of the tracer equation including total advection of tracers, horizontal and vertical diffusion of tracers, and surface boundary conditions. The total advection of the tracers is equal to the sum of the flux through the west face of the \( T \) box, plus the flux through the top of the \( T \) box, plus the zonal, meridional, and vertical flux divergences.

Implementation of the following equation is required for computing water temperature:

\[
\frac{\partial T}{\partial t} + \nabla \cdot ( \mu T ) = -K_H \nabla_H^2 T + K_Z \frac{\partial^2 T}{\partial z^2} - \frac{f(0)(1 - A) \delta(z) R_0 \theta(T - T_f)}{Z_{\text{mix}}} - R_i(T - T_0),
\]

(3)

where

\( \frac{\partial T}{\partial t} \) = the time rate of change of temperature,

\( \nabla \cdot ( \mu T ) \) = heat advected into the \( T \) box,

\( K_H \nabla_H^2 T \) = horizontal diffusion,

\( K_Z \frac{\partial^2 T}{\partial z^2} \) = vertical diffusion,

\( f(0) \) = the ice growth/melting rate in open water,

\( (1 - A) \) = the percentage of the open water area,

\( \delta(z) \) = 1 in the mixed layer and 0 otherwise,

\( R_0 \) = the ratio of the latent heat of fusion of sea ice to the heat capacity of water,

\( \theta(T - T_f) \) = 1 when the mixed-layer temperature \( T \) is greater than the freezing point \( T_f \) and 0 otherwise,
\( Z_{\text{mix}} \) = the mixed-layer thickness,
\( R_t \) = the robust constraint for water temperature, and
\( T_0 \) = the Levitus climatology temperature.

Similarly, implementation of the following equation is required for computing salinity:

\[
\frac{\partial S}{\partial t} + \nabla \cdot (uS) = K_H \nabla^2 H S + K_Z \frac{\partial^2 S}{\partial z^2} - \frac{0.035G_h \delta(z)}{Z_{\text{mix}}} - R_t \left( S - S_0 \right),
\]  

(4)

where
\( \frac{\partial S}{\partial t} \) = the time rate of change of salinity,
\( \nabla \cdot (uS) \) = salinity advected into the T box,
\( K_H \nabla^2 H \) = horizontal diffusion,
\( K_Z \frac{\partial^2 S}{\partial z^2} \) = vertical diffusion,
\( G_h \) = the total growth rate of open water and sea ice,
\( \delta(z) \) = 1 in the mixed layer and 0 otherwise,
\( Z_{\text{mix}} \) = the mixed-layer thickness,
\( R_t \) = the robust constraint for water temperature, and
\( S_0 \) = the Levitus climatological salinity.

2.2.4.4 CSC Compute External Mode

CSC Compute External Mode must employ sequential overrelaxation to solve the LaPlacian equation for the external mode of velocity in terms of a mass transport stream function. To provide predictions for a model basin with islands, CSC Compute External Mode must utilize the method of "hole relaxation." The vorticity driving function computed by CSC Compute Internal Mode and Vorticity Driving Function is required as input to CSC Compute External Mode.

CSC Compute External Mode must read in the relaxation solutions from the previous two timesteps and extrapolate the solutions forward in time for use as an initial guess for the relaxation solution of the current timestep. This minimizes the number of scans required for convergence to the solution.

2.2.4.5 CSC Direct Access Manager

CSC Direct Access Manager is included in ocean to perform input/output (I/O) tasks on slab and slab incidental data. The I/O may be real, when data is actually stored on disk, or simulated, when all data is resident in memory (core-contained). Either way, CSC Direct Access Manager insulates the rest of the model from any dependence on the type of I/O system actually used. When the ocean model is configured to run in a core-contained mode, CSC Direct Access Manager contains routines to initialize a virtual disk from tape, transfer data to memory and vice versa, and save a virtual disk to tape.
The I/O system must satisfy two main criteria. First, a complete record of all prognostic variables must reside on permanent disk at the end of each timestep. This allows restarting the model for continuous daily execution. Secondly, the I/O system must feed data to and from memory in a row-by-row manner. Data for one row including all east-west and vertical gridpoints is termed slab data. At any one time, only the slab data necessary for the computation should be present. This includes data for the previous, the current, and the next timestep.

Three disk units are needed for the I/O system: one for the \( N \) timestep data, one for the \( N - 1 \) timestep data, and one for the newly computed \( N + 1 \) timestep data. Disk units are permuted between units 13, 14, and 15 to minimize data transfers. Primary slab data including tracer data and \( U \) and \( V \) velocities are managed as follows: On timestep 1, \( N - 1 \) and \( N \) data will be read from units 13 and 14, and the \( N + 1 \) data will be written to 15. On timestep 2, units 14 and 15 will be read and the \( N + 1 \) data will be written to unit 13. On timestep 3, 15 and 13 will be read and the \( N + 1 \) data will go to 14, etc. In a core-contained mode, the file I/O described above is simulated by CSC Direct Access Manager.

CSC Direct Access Manager also controls the I/O for the slab incidental data, such as the number of vertical levels of the ocean at \( T \) and \( U,V \) points. Because these variables are constant in time, there is no need to keep multiple records of them. Slab incidental data are managed in a manner similar to the primary slab data with some considerations. The method is outlined in Sec. 3.4.6. CSC Direct Access Manager also manages the storage and retrieval of the two-dimensional horizontal fields and the timestep counter, total elapsed time, and the area and volume of the basin.

2.2.5 CSC Next Timestep

CSC Next Timestep is used after each timestep run of CSCs Ice and Ocean to update timestep information. It calculates new timestep information including date-time group, new currents, and new mixed-layer temperatures for the next pass through the ice and ocean models. Positions of previous and present fields are swapped.

2.2.6 CSC Output

The CSC Output is the last CSC executed in PIPS2.0. It is responsible for writing the data files containing the ice model and ocean model restart data. These restart values are used by PIPS2.0 for the next model run.

3.0 DETAILED DESIGN

This section gives design information for each PIPS2.0 CSC described in Sec. 2.0. Each CSC is divided into CSUs.

3.1 CSC Program Driver

CSC Program Driver serves as the main driving routine for PIPS2.0. It must establish the proper data in memory needed to restart the model each day, drive the Ice and Ocean CSCs, and save the data needed for the next model run.
3.1.1 CSU DRIVER

CSU DRIVER serves as the program driver for CSCI PIPS2.0.

3.1.1.1 CSU DRIVER Design Specification/Constraints

There are no known constraints.

3.1.1.2 CSU DRIVER Design

Input Data Elements:
- FCORSP: array of sine of latitude positions for each gridpoint
- IDTG: date-time group in the form YYMMDDHH
- ITSTEP: number of timesteps for run
- PLTSTP: interval in timesteps at which to plot
- PRTSTP: interval in timesteps at which to print
- IRIVER: x component of river discharge
- JRIVER: y component of river discharge
- KRIVER: z component of river discharge
- RIVER: river discharge temperature
- SMIX: monthly interpolated Levitus salinity fields
- TMIX: monthly interpolated Levitus temperature fields

Output Data Elements:
- GAIHX: x component of geostrophic wind
- GAIHY: y component of geostrophic wind
- HEFF: mean ice thickness per grid cell
- UICE: x component of ice drift
- VICE: y component of ice drift
- AREA1: fraction of grid cell covered with ice

/FCORSP/
- FCORSP: array of sine of latitude positions for each gridpoint

/FULLWD/
- NNRENY: final timestep to compute for the current run

/LEVITUS/
- SMIX: monthly interpolated Levitus salinity fields
- TMIX: monthly interpolated Levitus temperature fields

/MAST/
- HEFFM: land/sea mask for thermodynamic variables
- OUT: land/sea mask including outflow conditions for thermodynamic variables
- UVM: land/sea mask for velocity variables

/RFOR 2/
- GWATX: x component of the ocean current
- GWATY: y component of the ocean current
/RSTRT/
  AREA1  fraction of grid cell covered with ice
  HEFF   mean ice thickness per grid cell
  TICE   mixed-layer temperature if open water; ice temperature if ice cover
  UICEC  intermediate x component of ice drift
  VICEC  intermediate y component of ice drift
  UICE   x component of ice drift
  VICE   y component of ice drift

/RVR/
  IRIVER  x component of river discharge
  JRIVER  y component of river discharge
  KRIIVER z component of river discharge
  RIVER  river discharge temperature

/TSTEP/
  IDTG    date-time group in the form YYMMDDHH
  IRSTRT  if 0, restart from previous run; otherwise, restart from constant conditions
  ITSTEP  number of timesteps for run
  MDY     day calculated from inputted data
  MHR     hour calculated from inputted data
  MM      month calculated from inputted data
  MYR     year calculated from inputted data
  PLTSTP  interval in timesteps at which to write output
  PRTSTP  interval in timesteps at which to plot output

Parameters:
stored in file ice.par
  IMTM1  IMT-1; = 359
  JMTM1  JMT-1; = 359
  IMT    total number of T grid boxes zonally; = 360
  JMT    total number of T grid boxes meridionally; = 360
  IMTP1  IMT+1; = 361
  JMTP1  JMT+1; = 361
  NXM1   IMT-2; = 358
  NYM1   JMT-2; = 358
  MIDY   80
  IRV    80
  LSEG   maximum number of sets of start and end indices; = 10
  NISLE  number of islands in model; = 4
  NT     number of tracer type variables carried in the model; = 2
  KM     total number of vertical levels; = 15

Data element design information is provided in Sec. 4.0

Local Data Elements:
  I       index counter  integer
  IJUL    Julian date     integer
  ILA     index of loop over rows/latitudes integer
  ILO     index of loop over columns/longitude integer
  J       index counter    integer
K
MMI
NRV
RADI

index counter
index for the Julian date of the previous month
index counter
degrees per radian

integer
integer
integer
integer

Logic Flow:

Upon start of the program, several inputs are read from standard input. These inputs include the
number of timesteps for the run (ITSTEP), the interval in timesteps at which to plot the output
(PLTSTP), the interval in timesteps at which to print the output (PRTSTP), the type of initialization
(IRSTRT), and the date-time group (IDTG). The DAYNUM CSU is called to compute the year
(MYR), month (MM), day (MDY), and hour (MHR) from the input date-time group. From the
month, day, and hour, the final timestep to compute for the current run is calculated.

The BNDRY CSU is called to read the velocity, thermodynamic, and outflow land/sea boundary
parameters from a file. The latitude positions of the gridpoints are read from a file and are then
converted from degrees to radians. The values of IRIVER, JRIVER, KRIVER, and RIVER are also
read from a file for every IRV. The ice and ocean restart values are read by calling the RST_OCN
and RST_ICE CSUs. The Levitus temperature and salinity fields (TMIX and SMIX) are read from
input files.

After all input data are read, the model is executed by running the CSC Ice and CSC Ocean
portions of PIPS2.0 for each timestep. For each timestep, CSC Next Timestep is called to update
the new timestep, moving previous timestep fields back one timestep and moving present timestep
fields into previous timestep fields.

Once all timesteps have been executed, the ice and ocean model restart data are written to files
by calling the SAV_ICE and SAV_OCN CSUs. A graphics file of output fields is also written
directly from CSU DRIVER. These values include the x component of the geostrophic wind (GAILX),
y component of the geostrophic wind (GAINY), the mean ice thickness per grid cell (HEFF), the
x component of the ice drift (UICE), y component of the ice drift (VICE), and the fraction of grid
cell covered with ice (AREA1).

Local Data Files:

The CSU Driver reads data from five different data files. The file connected to logical unit 14
is named newlatu.dat and is a permanent, IEEE binary file containing the latitude positions of
gridpoints. The river rate discharge data is in the file named river_XXX.dat where XXX is the month;
this file is a permanent, ASCII file connected to logical unit 19. The files named for018_tu_xx.dat
and for018_su_xx.dat, where XX is the month number, are connected to logical units 10 and 11,
respectively. Both files are permanent, IEEE binary files. The “tu” file contains the interpolated
Levitus monthly ocean temperature and the “su” file contains the interpolated Levitus monthly
ocean salinity values. The final file contains the ice model results and is named 92xxx.dat where
XXXX is the month and day. This file is also a permanent, binary file and is connected to logical
unit 31.

3.2 CSC Input

CSC Input is used for reading and/or calculating inputs necessary for operation of PIPS2.0.
Four CSUs are called to perform the functions required by the CSC Input.

DAYNUM converts date-time group
BNDRY reads the boundary data
RST_ICE reads ice model restart data written in the previous run of the model
RST_OCN reads ocean model restart data written in the previous run of the model

3.2.1 CSU DAYNUM

The DAYNUM CSU converts the date-time group in the form YYMMDDHH to the year, month, day, and hour.

3.2.1.1 CSU DAYNUM Design Specification/Constraints

There are no known constraints.

3.2.1.2 CSU DAYNUM Design

Input Data Elements:

/TSTEP/
  IDTG  date-time group in the form YYMMDDHH

Output Data Elements:

/TSTEP/
  MDY    day from the date-time group
  MHR    hour from the date-time group
  MM     month from the date-time group
  MYR    year from the date-time group

Data element design information is provided in Sec. 4.0

Local Data Elements:

IDTG1    temporary variable used in determining the day    integer
IDTG2    temporary variable used in determining the hour    integer

Algorithms:

To calculate the year (MYR), day (MDY), and hour (MHR), use the following:

\[ MYR = \frac{IDTG}{1000000} \]
\[ MM = \frac{(IDTG - (MYR \times 1000000))}{10000} \]
\[ IDTG1 = (IDTG/100000) \times 10000 \]
\[ MDY = \frac{(IDTG - IDTG1)}{100} \]
\[ IDTG2 = (IDTG/100) \times 100 \]
\[ MHR = IDTG - IDTG2 \]

where IDTG is the date-time group in the form YYMMDDHH. The date-time group IDTG is input to the CSU and the year, day, and hour are output.

Logic Flow:

The DAYNUM CSU is called by the ICEMDL CSU after it reads the date-time group and also after it increments the date-time group for the next timestep. The year, month, day, and hour are
calculated from the date-time group and the values are returned to the calling CSU. The month is then used in determining the snowfall rate.

3.2.2 CSU BNDRY

CSU BNDRY reads the velocity, thermodynamic, and outflow land/sea boundary parameters from a file.

3.2.2.1 CSU BNDRY Design Specification/Constraints

Land/sea boundary parameters are read from a file connected to logical unit 15.

3.2.2.2 CSU BNDRY Design

Input Data Elements:

| UVM       | land/sea mask for velocity variables |
| OUT       | land/sea mask including outflow conditions for thermodynamic variables |
| HEFFM     | land/sea mask for thermodynamic variables |

Output Data Elements:

/MASK/

| UVM       | land/sea mask for velocity variables |
| OUT       | land/sea mask including outflow conditions for thermodynamic variables |
| HEFFM     | land/sea mask for thermodynamic variables |

Parameters:

stored in file ice.par

The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Logic Flow:

The BNDRY CSU is called by CSU DRIVER to define all boundaries. The land/sea masks for the maximum number of velocity fields, thermodynamic fields, and outflow fields are read from a file connected to logical unit 15.

The land/sea masks for the thermodynamic, velocity, and outflow fields are returned to the CSU DRIVER.

Local Data Files:

The file connected to logical unit 15 is named mask_u.dat and is a permanent, binary file containing the land/sea tables that define the PIPS2.0 model. The file contains the land/sea masks for the thermodynamic fields, velocity fields, and outflow grid cells.

3.2.3 CSU RST_ICE

The RST_ICE CSU reads the ice restart data written from the previous ice model run.

3.2.3.1 CSU RST_ICE Design Specification/Constraints

Restart values are read from a file connected to logical unit 16.
3.2.3.2 CSU RST\_ICE Design

Input/Output Data Elements:
- AREA1: fraction of the grid cell covered by ice
- HEFF: mean ice thickness per grid cell
- TICE: mixed-layer temperature for open water or ice temperature for ice cover
- UICE: x component of the ice drift
- UIPEC: intermediate x component of the ice drift
- VICE: y component of the ice drift
- VICEC: intermediate y component of the ice drift

Parameters:
stored in file ice.par

The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Logic Flow:
The x and y components of the ice drift (UICE and VICE), intermediate x and y components of the ice drift (UIPEC and VICEC), mean ice thickness (HEFF), fraction of the grid cell covered by ice (AREA1), and mixed-layer temperature for open water or ice temperature for ice cover (TICE) are read from the permanent binary restart data file. All parameters are returned to the calling CSU.

This CSU is implemented by the CSU DRIVER.

Local Data Files:
The restart data are read from 92<mmdd>.res (unit 16), where <mmdd> is the two-digit month, and two-digit day of the previous day. This permanent, binary file contains the unformatted restart data from the previous day. The x and y components of the ice drift, the intermediate x and y components of the ice drift used in semi-implicit timestepping, the mean ice thickness per grid cell, the fraction of the grid cell covered by thick ice, the negative ice to be melted, and the mixed-layer temperature (in the case of open water) or the ice temperature (in the case of an ice cover) are used by the ocean model run for the current day and the ice model run for the next day.

3.2.4 CSU RST\_OCN

The RST\_OCN CSU reads the ocean model restart data written from the previous PIPS2.0 model run.

3.2.4.1 CSU RST\_OCN Design Specification/Constraints

Restart values are read from a file connected to logical unit 13.

3.2.4.2 CSU RST\_OCN Design

Input Data Elements:
- FW1: heat above the freezing temperature
- GICE: ice growth rate computed by the ice model
- TM1: previous timestep mixed-layer temperature
- TM2: previous two timesteps mixed-layer temperature
Software Test Description for PIPS2.0

TMP  array containing previous timestep temperature and salinity values
UTEMP  previous timestep x component of ocean current
VTEMP  previous timestep y component of ocean current

**Output Data Elements:**
GWATX  x component of geostrophic ocean current
GWATY  y component of geostrophic ocean current

/COX2/
FW1  heat above the freezing temperature
GICE  ice growth rate computed by the ice model
SHICE  total ice thickness
TFRZ  temperature at freezing point
TM1  previous timestep mixed-layer temperature
TM2  previous two timesteps mixed-layer temperature
TMP  array containing previous timestep temperature and salinity values
UTEMP  previous timestep x component of ocean current
VTEMP  previous timestep y component of ocean current

/CURNTS/
T1  previous timestep of temperature and salinity
T2  previous two timesteps of temperature and salinity
U1  previous timestep of x component of ocean current
V1  previous timestep of y component of ocean current

/FULLWD/
ITT  timestep counter

/OCEANS/
FW  oceanic heat flux

**Parameters:**
stored in file ice.par

The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

**Local Data Elements:**
CP  constant of water heat capacity  real
EXCHNG  number of times to exchange ice/ocean data  real
I  index counter  integer
J  index counter  integer
J1  temporary variable  integer
M  index counter  integer
RLATNT  constant of water latent heat  real
TDAY  timestep to exchange ice/ocean data  real
ZMIX  depth of mixed layer (30 m)  real

**Logic Flow:**
The x and y components of the currents, temperature, and the salinity from the ocean model are read from the permanent binary restart data file. Input currents are in centimeters per second,
temperatures are in degrees Celsius, and salinities are calculated from 0.035 – salinity to find the
difference. Currents are converted to m/s and mixed-layer temperatures are converted to Kelvins.
The seawater freezing temperature is calculated according to salinity as \(-54.4 \times \text{salinity}\). The u and
v components of currents, ocean temperatures, and salinity for level 1 from the last two timesteps
are stored.

The ice thickness growth rate of open water, the total ice thickness growth rate, and the heat
above the freezing temperature from the previous timestep are read from the file. If the growth rate
is greater than that of the heat above freezing temperature, and the heat above freezing is >0, the
growth rate is set to 0 (no more cooling). Lastly, the heat flux is calculated.

**Algorithms:**
The oceanic heat flux is calculated using:

\[
Q = c_p \ z_m \ (T_{m_i-1})/t_d n_i + GL_w, \tag{5}
\]

where \(c_p\) is the constant of water heat capacity,

\(z_m\) is the depth of the mixed layer,

\(T_{m_i}\) is the mixed-layer temperature from the previous timestep,

\(T_{m_i-1}\) is the mixed-layer temperature from two timesteps ago,

\(t_d n_i\) is the timestep to exchange data on \((t_d\) is the number of seconds in a day and

\(n_i\) is the number of times to exchange data),

\(G\) is the ice growth rate, and

\(L_w\) is the constant of water latent heat.

CSURST_OCN is implemented by CSU DRIVER.

**Local Data Files:**
The restart data are read from for010_<mmd><dd>.dat (unit 13), where \(<mmd><dd>\) is the two-digit
month and two-digit day of the previous day. This permanent, binary file contains the unformatted
restart data from the previous day.

3.3 CSC Compute Ice
3.3.1 CSC Compute Ice Driver

3.3.1.1 CSU ICEMDL

CSU ICEMDL is the main driving subroutine for the Hibler viscous-plastic sea ice model
(1979; 1980). Minor modifications were made to allow the original routine to function as a subroutine
in the CSCI PIPS2.0.

3.3.1.1.1 CSU ICEMDL Design Specification/Constraints – The snowfall rates are based on
monthly climatological values (Maykut and Untersteiner 1969; Parkinson and Washington 1979).
There are no known constraints.

3.3.1.1.2 CSU ICEMDL Design

**Input Data Elements:**

- AMASS ice mass per grid area
- DELTAT timestep (s)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTAX</td>
<td>x (longitude) grid spacing (deg)</td>
</tr>
<tr>
<td>DELTAY</td>
<td>y (latitude) grid spacing (deg)</td>
</tr>
<tr>
<td>DRAGA</td>
<td>asymmetric water drag plus the Coriolis parameter</td>
</tr>
<tr>
<td>DRAGS</td>
<td>symmetric water drag</td>
</tr>
<tr>
<td>ETA</td>
<td>nonlinear shear viscosity</td>
</tr>
<tr>
<td>FO</td>
<td>growth rate of thin ice</td>
</tr>
<tr>
<td>GAREA</td>
<td>change of compactness due to freezing or melting</td>
</tr>
<tr>
<td>HCORR</td>
<td>additional ice to be melted for mixed-layer balance</td>
</tr>
<tr>
<td>HDIFF1</td>
<td>net rate of total open water growth</td>
</tr>
<tr>
<td>ZETA</td>
<td>nonlinear bulk viscosity</td>
</tr>
</tbody>
</table>

/GROW/
FHEFF total growth rate of thick ice

/MASK/
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEFFM</td>
<td>land/sea mask for thermodynamic variables</td>
</tr>
<tr>
<td>OUT</td>
<td>land/sea mask including outflow conditions for thermodynamic variables</td>
</tr>
<tr>
<td>UVM</td>
<td>land/sea mask for velocity variables</td>
</tr>
</tbody>
</table>

/RFOR2/
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAIRX</td>
<td>x component of the geostrophic wind</td>
</tr>
<tr>
<td>GAIRY</td>
<td>y component of the geostrophic wind</td>
</tr>
<tr>
<td>GWATX</td>
<td>x component of the ocean current</td>
</tr>
<tr>
<td>GWATY</td>
<td>y component of the ocean current</td>
</tr>
</tbody>
</table>

/RSTRT/
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA1</td>
<td>fraction of grid cell covered by ice</td>
</tr>
<tr>
<td>HEFF</td>
<td>mean ice thickness per grid cell</td>
</tr>
<tr>
<td>UICE</td>
<td>x component of ice drift</td>
</tr>
<tr>
<td>UICEC</td>
<td>intermediate x component of the ice drift</td>
</tr>
<tr>
<td>VICE</td>
<td>y component of ice drift</td>
</tr>
<tr>
<td>VICEC</td>
<td>intermediate y component of the ice drift</td>
</tr>
</tbody>
</table>

/TSTEP/
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDTG</td>
<td>date-time group; read from standard input (YYMMDDHH)</td>
</tr>
<tr>
<td>ITSTEP</td>
<td>number of timesteps for run</td>
</tr>
<tr>
<td>MDY</td>
<td>day calculated from date-time group</td>
</tr>
<tr>
<td>MM</td>
<td>month calculated from date-time group</td>
</tr>
<tr>
<td>PRTSTP</td>
<td>interval in timesteps at which to write results</td>
</tr>
</tbody>
</table>

Output Data Elements:
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A22</td>
<td>minimal compactness allowed</td>
</tr>
<tr>
<td>AMASS</td>
<td>ice mass per grid area</td>
</tr>
<tr>
<td>DIFF1</td>
<td>harmonic diffusion constant</td>
</tr>
<tr>
<td>DRAGA</td>
<td>asymmetric water drag plus the Coriolis parameter</td>
</tr>
<tr>
<td>DRAGS</td>
<td>symmetric water drag</td>
</tr>
<tr>
<td>ERROR</td>
<td>maximum error allowed in the relaxation scheme</td>
</tr>
<tr>
<td>ETA</td>
<td>nonlinear shear viscosity</td>
</tr>
<tr>
<td>FO</td>
<td>growth rate of thin ice</td>
</tr>
<tr>
<td>GAREA</td>
<td>change of compactness due to freezing or melting</td>
</tr>
<tr>
<td>HCORR</td>
<td>additional ice to be melted for mixed-layer balance</td>
</tr>
</tbody>
</table>
HDIF1: net rate of total open water growth
HO: demarcation between thick and thin ice
LAD: determines the time-stepping scheme used
THETA: indicates a backwards timestep
ZETA: nonlinear bulk viscosity

/GROW/
FHEFF: total growth rate of thick ice

/RFOR2/
GWATX: x component of the ocean current
GWATY: y component of the ocean current

/RESTRT/
AREA1: fraction of grid cell covered by ice
HEFF: mean ice thickness per grid cell
UICE: x component of ice drift
UICEC: intermediate x component of ice drift
VICE: y component of ice drift
VICEC: intermediate y component of ice drift

/SNOW/
SNRT: snowfall rate

/STEP/
DELTAT: timestep (s)
DELTAX: x (longitude) grid spacing (deg)
DELTAY: y (latitude) grid spacing (deg)

/STEPSP/
DELTXA: x (longitude) grid spacing for thermodynamic fields (m)
DELTXU: x (longitude) grid spacing for velocity fields (m)
DELTYA: y (latitude) grid spacing for thermodynamic fields (m)
DELTYU: y (latitude) grid spacing for velocity fields (m)

Parameters:
stored in file ice.par
The ice.par parameters are described in Sec. 3.1.1.2.
Data element design information is provided in Sec. 4.0.

Local Data Elements:
ARSUM: sum of ice compactness over all grid squares
ARSUM1: net ice concentration
DELT: 1/2 of the timestep (s)
FHSUM: sum of growth rate of ice over all grid squares
FHSUM1: net ice growth
GRSUM: sum of thin ice over all grid squares
GRSUM1: net open water growth
I: index counter
ICOUNT: timestep counter
IMT     total number of T grid boxes zonally (360)     integer
IMTM1   total number of T grid boxes zonally – 1 (359) integer
IT      number of timesteps allowed during run       integer
ITSTEP  number of timesteps for run                 integer
J       index counter                                integer
JMT     total number of T grid boxes meridionally (360) integer
JMTM1   total number of T grid boxes meridionally – 1 (359) integer
KSTEP   counter for timesteps                        integer
MIDY    (160)                                        integer
PRTSTEP  the interval, in timesteps, at which to print results real
Radian  degrees per radian (57.29578°)               real
RADIUS  radius of the earth (6370.0 × 10^3 m)         real
SM      maximum value of SMU and SMV                  real
SMU     maximum value of x component of the velocity  real
difference between times T+1 and T
SMV     maximum value of y component of the velocity  real
difference between times T+1 and T
SQ      squared velocity                              real
SQ1     squared velocity difference between times t+1 and t real
THEFF   total basin ice thickness                     real
THEFF1  total basin ice thickness after outflow adjustment real
THEFF2  sum of thickness over all grid squares        real
TOUT    net outflow                                  real
TOUT1   outflow for the current timestep              real
UERR    x component velocity difference between time t+1 and t real
VERR    y component velocity difference between time t+1 and t real

Data Conversions:
When calculating the grid spacing in meters from the grid spacing in degrees, degrees must be converted to radians.

Logic Flow:
Before solving any equations, initial conditions for the model are defined. First, the latitude grid spacing in meters for the velocity (DELTYU) and thermodynamic (DELTYA) fields are defined. These values are assumed to be equal and are constant for all latitudes:

\[
\text{DELTYU} = \text{DELTAY} \times \frac{\text{RADIUS}}{\text{RADIUS/RADIAN}}
\]
\[
\text{DELTYA} = \text{DELTAY} \times \frac{\text{RADIUS}}{\text{RADIUS/RADIAN}},
\]

where DELTAY is the y (latitude) grid spacing in degrees (0.28575°), RADIUS is the radius of the Earth (6370.0 × 10^3 m), and RADIUSRadian is degrees per radian (57.29578°).

Next, the grid length in meters for the ice-drift velocity (DELTXU) and the grid length for ice thickness and concentration (DELTXA) are defined:

\[
\text{DELTXU}(J) = \cos(\text{DELTAY} \times (\text{FLOAT}(J) - (\text{MIDY} - 0.5) - 0.5)/\text{RADIUS}) \times \text{DELTAX} \times \frac{\text{RADIUS}}{\text{RADIUS/RADIAN}}
\]
\[
\text{DELTXA}(J) = \cos(\text{DELTAY} \times (\text{FLOAT}(J) - \text{MIDY} - 0.5)/\text{RADIUS}) \times \text{DELTAX} \times \frac{\text{RADIUS}}{\text{RADIUS/RADIAN}}
\]
where DELTAY is the y (latitude) grid spacing in degrees, \( J \) is the index of the number of T grid boxes meridionally and DELTAX is the x (longitude) grid spacing in degrees (0.28575\(^\circ\)), RADIUS is the radius of the Earth, and RADIANS is degrees per radian.

The month and day are used to initialize the snowfall rate based on monthly climatological values (Parkinson and Washington 1979).

<table>
<thead>
<tr>
<th>Month</th>
<th>Snowfall Rate (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>(3.215 \times 10^{-9})</td>
</tr>
<tr>
<td>February</td>
<td>(3.215 \times 10^{-9})</td>
</tr>
<tr>
<td>March</td>
<td>(3.215 \times 10^{-9})</td>
</tr>
<tr>
<td>April</td>
<td>(3.215 \times 10^{-9})</td>
</tr>
<tr>
<td>May</td>
<td>(19.29 \times 10^{-9})</td>
</tr>
<tr>
<td>June</td>
<td>0</td>
</tr>
<tr>
<td>July</td>
<td>0</td>
</tr>
<tr>
<td>August 1–19</td>
<td>0</td>
</tr>
<tr>
<td>August 20–31</td>
<td>(49.603 \times 10^{-9})</td>
</tr>
<tr>
<td>September</td>
<td>(49.603 \times 10^{-9})</td>
</tr>
<tr>
<td>October</td>
<td>(49.603 \times 10^{-9})</td>
</tr>
<tr>
<td>November</td>
<td>(3.215 \times 10^{-9})</td>
</tr>
<tr>
<td>December</td>
<td>(3.215 \times 10^{-9})</td>
</tr>
</tbody>
</table>

Additional initial values are calculated. The maximum error allowed in the relation scheme (ERROR) is set to:

\[
\text{ERROR} = 0.000001 \times 5.0.
\]

DIFF1, the harmonic diffusion constant, is redefined in spherical coordinates. It is assumed to be a constant coefficient for the harmonic function of the diffusion for ice concentration and thickness at all locations. Note that the harmonic term coefficient is defined to be 0.004 times the grid length to have the diffusion approximate to the change of ice thickness and concentration over one timestep:

\[
\text{DIFF1} = 0.004 \times \text{DELTXA},
\]

where DELTXA is the x (longitude) grid spacing in meters for thermodynamic fields. This term will be used later in determining the advection of the ice thickness and compactness.

To calculate the growth rates of the ice, the demarcation between thick and thin ice is defined as \(\text{HO} = 0.5 \times 2.0\).

CSU XSUM is called to calculate the total basin ice thickness except at outflow boundary gridpoints.

The initial value of THETA is set to 1.0. This is used later in the RELAX CSU to indicate a backward timestep.

CSU FORM is called to calculate the basic parameters for the RELAX CSU. The ice drift, ocean currents, marine winds, outflow, thermodynamic land/sea mask, ice thickness, and ice...
concentration are used to determine the average ice mass in a grid square, the symmetric and asymmetric water drags, the forcing fields, the ice pressure, and viscosities.

The average ice mass per grid square (AMASS), the nonlinear bulk viscosity (ZETA), and the nonlinear shear viscosity (ETA) are defined for each velocity field:

\[
\begin{align*}
\text{AMASS}(I,J) &= 0, \\
\text{ZETA}(I,J) &= \text{HEFF}(I,J,1) * 10^{11}, \\
\text{ETA}(I,J) &= \text{ZETA}(I,J)/4.0,
\end{align*}
\]

where HEFF is the mean ice thickness per grid cell.

The ice momentum balance equation is solved by calling the RELAX CSU. The ice advection term is also computed there.

The standard predictor corrector iteration scheme is used to center the nonlinear terms. The process begins by looping over the number of timesteps. First a prediction is done followed by a regular timestep. For the prediction, the third time level value for each x and y component of the ice drift is set to the value at the first time level. The intermediate x and y components of the ice drift are set to the x and y components of the ice drift at the first time level:

\[
\begin{align*}
\text{UICE}(I,J,3) &= \text{UICE}(I,J,1) \\
\text{VICE}(I,J,3) &= \text{VICE}(I,J,1) \\
\text{UICEC}(I,J) &= \text{UICE}(I,J,1) \\
\text{VICEC}(I,J) &= \text{VICE}(I,J,1)
\end{align*}
\]

The indicator for the backward timestep THETA is set to 1.0. The FORM CSU is called to calculate the terms for the momentum equation and then the equation is solved by calling the RELAX CSU. This relaxation solution is required to center the nonlinear terms.

After a prediction, a regular timestep is calculated. The first and second time levels of the x and y components of the ice drift are averaged:

\[
\begin{align*}
\text{UICE}(I,J,1) &= 0.5 * (\text{UICE}(I,J,1) + \text{UICE}(I,J,2)) \\
\text{VICE}(I,J,1) &= 0.5 * (\text{VICE}(I,J,1) + \text{VICE}(I,J,2))
\end{align*}
\]

The FORM CSU is called to calculate the terms for the momentum equation. The third time level value for each x and y component of the ice drift is set to the value at the first time level. The intermediate x and y components of the ice drift are set to the x and y components of the ice drift at the first time level. The first time level for each x and y component of the ice drift is set to the value of the second time level:

\[
\begin{align*}
\text{UICE}(I,J,3) &= \text{UICE}(I,J,1) \\
\text{VICE}(I,J,3) &= \text{VICE}(I,J,1) \\
\text{UICEC}(I,J) &= \text{UICE}(I,J,1) \\
\text{VICEC}(I,J) &= \text{VICE}(I,J,1) \\
\text{UICE}(I,J,1) &= \text{UICE}(I,J,2) \\
\text{VICE}(I,J,1) &= \text{VICE}(I,J,2)
\end{align*}
\]
The momentum equation is solved by calling the RELAX CSU. This relaxation solution is required to advance to the next timestep.

The \( t + 1 \) time values of UICE and VICE are saved for use in the advection of the ice thickness and compactness.

\[
\text{UICEC}(I,J) = \text{UICEC}(I,J,1), \\
\text{VICEC}(I,J) = \text{VICEC}(I,J,1),
\]

where \( I \) and \( J \) are loop indices for the total number of \( T \) grid boxes zonally and meridionally, respectively.

The squared velocity and squared velocity difference between times \( t+1 \) and \( t \) are calculated. This velocity information is then written to standard output along with the timestep counter and date-time group.

\[
\text{SQ} = \text{SQ} + \text{UICE}(I,J,1) ** 2 + \text{VICE}(I,J,1) ** 2 \\
\text{UERR} = \text{UICE}(I,J,1) - \text{UICE}(I,J,2) \\
\text{VERR} = \text{VICE}(I,J,1) - \text{VICE}(I,J,2) \\
\text{SQ1} = \text{SQ1} + (\text{UERR} * \text{UERR}) + (\text{VERR} * \text{VERR}) \\
\text{SMU} = \max(\text{abs}(\text{UERR},\text{SMU})) \\
\text{SMV} = \max(\text{abs}(\text{VERR},\text{SMV}))
\]

CSU ADVECT (Sec. 3.3.5.1) is called twice—once to compute the advection terms for the continuity equation of ice thickness and once for the advection of ice compactness. The advection is a divergence of ice-drift velocity times the ice thickness or ice concentration.

The forcing fields are used by the HEAT CSU (Sec. 3.3.7.1) to calculate the terms necessary for determining the heat balance. Next the GROWTH CSU (Sec. 3.3.5.3) is called to calculate the changes in ice thickness and ice concentration. The total ice in the basin, excluding outflow cells, is recalculated by calling CSU XSUM (Sec. 3.3.6.1).

Several sums are computed for ensuring conservation and monitoring various contributions to the ice changes. For each thermodynamic field, the mean ice thickness per grid cell (HEFF), the fraction of the grid cell covered by thick ice (AREA1), the total growth rate of thick ice (FHEFF), the change in compactness due to freezing or melting (GAREA), additional ice to be melted for mixed-layer balance (HCORR), and the net rate of total open water growth (HDIFF1) are corrected at the outflow grid cells:

\[
\text{HEFF}(I,J,1) = \text{HEFF}(I,J,1) * \text{OUT}(I,J) \\
\text{HEFF}(I,J,2) = \text{HEFF}(I,J,2) * \text{OUT}(I,J) \\
\text{HEFF}(I,J,3) = \text{HEFF}(I,J,3) * \text{OUT}(I,J) \\
\text{AREA1}(I,J,1) = \text{AREA1}(I,J,1) * \text{OUT}(I,J) \\
\text{AREA1}(I,J,2) = \text{AREA1}(I,J,2) * \text{OUT}(I,J) \\
\text{AREA1}(I,J,3) = \text{AREA1}(I,J,3) * \text{OUT}(I,J) \\
\text{FHEFF}(I,J) = \text{FHEFF}(I,J) * \text{OUT}(I,J)
\]
GAREA(I,J) = GAREA(I,J) * OUT(I,J)
HCORR(I,J) = HCORR(I,J) * OUT(I,J)
HDIFF1(I,J) = HDIFF1(I,J) * OUT(I,J)

Also, the sum of the thickness over all grid squares (THEFF2), the sum of ice compactness over all grid squares (ARSUM), and the sum of growth rate of ice over all grid squares (FHSUM) is 0

GRSUM = GRSUM + HEFF1(I,J)
THEFF2 = THEFF2 + HEFF(I,J,1)
ARSUM = ARSUM + AREA1(I,J,1)
FHSUM = FHSUM + FHEFF(I,J)

Then the net values for all timesteps are determined. These include the net open water growth (GRSUM1), the net ice growth (FHSUM1), net ice concentration (ARSUM1), the outflow for the current timestep (TOUT1), the total basin ice thickness (THEFF), and the net outflow (TOUT):

GRSUM1 = GRSUM1 + GRSUM
FHSUM1 = FHSUM1 + FHSUM
ARSUM1 = ARSUM1 + ARSUM
TOUT1 = THEFF - THEFF2 - THEFF1
THEFF = THEFF2
TOUT = TOUT + TOUT1

If the current timestep interval is one at which to write to standard output, then the values of the date-time group (IDTG), the current timestep (ICOUNT), the squared velocity (THEFF), the squared velocity difference (THEFF1), the outflow for the current timestep (TOUT1), the net outflow (TOUT), ice growth for the current timestep (FHSUM), the net ice growth (FHSUM1), the sum of the thin ice over all grid squares (GRSUM), the net open water growth (GRSUM1), the sum of ice compactness over all grid squares (ARSUM), and the net ice concentration (ARSUM1) are printed.

If the current timestep is the last timestep, then the process is terminated. This iterative process continues until all timesteps are processed.

3.3.2 CSC Atmospheric Forcing

Atmospheric forcing fields are read in CSU RFORCE, which in turn calls CSU GEOWIND to calculate the geostrophic winds.

3.3.2.1 CSU RFORCE

CSU RFORCE reads the atmospheric forcing fields derived from the 1992 results of NOGAPS and calls CSU GEOWIND to calculate the geostrophic winds. Winds are output for use by the CSC Compute Ocean.

3.3.2.1.1 CSU RFORCE Design Specification/Constraints – Atmospheric forcing data produced by the NOGAPS model are read from a file connected to unit 12. There are no known constraints.
3.3.2.1.2 CSU RFORCE Design –

**Input Data Elements:**
- TA: NOGAPS atmospheric forcing – surface air temperature
- PSA: NOGAPS atmospheric forcing – surface air pressure
- ESA: NOGAPS atmospheric forcing – surface vapor pressure
- FSH1: NOGAPS atmospheric forcing – incoming solar radiation (short wave)
- PSB: NOGAPS atmospheric forcing – total heat flux
- ESB: NOGAPS atmospheric forcing – sensible heat flux

**Output Data Elements:**
- ES: NOGAPS atmospheric forcing – surface vapor pressure
- ES1: NOGAPS atmospheric forcing – sensible heat flux
- FSH: NOGAPS atmospheric forcing – incoming solar radiation (short wave)
- GAIRX: x component of the geostrophic wind
- GARY: y component of the geostrophic wind
- PS: NOGAPS atmospheric forcing – surface air pressure
- PS1: NOGAPS atmospheric forcing – total heat flux
- TAIR: NOGAPS atmospheric forcing – surface air temperature

**Parameters:**
*stored in file ice.par*

The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

**Local Data Elements:**
- ESA(IMT1,JMT1): NOGAPS atmospheric forcing – surface vapor pressure
- ESB(IMT1,JMT1): NOGAPS atmospheric forcing – sensible heat flux
- FSH1(IMT1,JMT1): NOGAPS atmospheric forcing – incoming solar radiation
- GAX(IMT1,JMT1): x component of the wind velocity
- GAY(IMT1,JMT1): y component of the wind velocity
- I: index counter
- IMT: total number of T grid boxes zonally (360)
- IMTP1: total number of T grid boxes zonally + 1 (361)
- J: index counter
- JMT: total number of T grid boxes meridionally (360)
- JMT1: total number of T grid boxes meridionally + 1 (361)
- PSA(IMT1,JMT1): NOGAPS atmospheric forcing – surface pressure
- PSB(IMT1,JMT1): NOGAPS atmospheric forcing – total heat flux
- TA(IMT1,JMT1): NOGAPS atmospheric forcing – air temperature

**Logic Flow:**
The RFORCE CSU is called by the ICEMDL CSU to read the forcing fields.

The atmospheric forcing fields are read from a file connected to logical unit 12. These include NOGAPS atmospheric forcing values for the air temperature (TA), the surface pressure (PSA), the surface vapor pressure (ESA), the incoming solar radiation (FSH1), the total heat flux (PSB), and the sensible heat flux (ESB).

The CSU GEOWIND is called to calculate the x and y components of the geostrophic wind from the surface pressure fields of atmospheric forcing.
Before returning to the calling CSU, the atmospheric fields are converted into units needed for the ice portion of the model.

**Local Data Files:**
The NOGAPS atmospheric forcing data are in a permanent, IEEE binary file called p92<mmdd>.dat (unit 12), where <mmdd> is the current two-digit month and two-digit day.

**3.3.2.2 CSU GEOWIND**
The GEOWIND CSU calculates geostrophic winds from surface pressure fields.

**3.3.2.2.1 CSU GEOWIND Design Specification/Constraints – There are no known constraints.**

**3.3.2.2.2 CSU GEOWIND Design**

**Input Data Elements:**
- **PS**
  - NOGAPS atmospheric forcing – surface air pressure
- **/CORSP/**
  - **FCORSP**
    - array of sine of latitude positions for each gridpoint
- **/STEPSP/**
  - **DELTXU**
    - x (longitude) grid spacing for velocity fields (m)
  - **DELYTUY**
    - y (latitude) grid spacing for velocity fields (m)

**Output Data Elements:**
- **G1X**
  - x component of geostrophic wind
- **G1Y**
  - y component of geostrophic wind

**Parameters:**
*stored in file ice.par*
The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

**Local Data Elements:**
- **CMBNM**
  - conversion factor from millibars to N/m² (100.0 N/m²)
  - real
- **F**
  - Coriolis parameter (1.46 × 10⁻⁴/s)
  - real
- **FPIN**
  - constant used in calculating the x and y components of the geostrophic wind (1.0 / (F * RHO))
  - real
- **I**
  - index counter
  - integer
- **J**
  - index counter
  - integer
- **RHO**
  - density of air (1.3 km/m³)
  - real

**Algorithms:**
To calculate the x (G1X) and y (G1Y) components of the geostrophic wind, the following equations are used:

\[
FPIN = 1.0 / (F \times RHO)
\]

\[
G1X_{i+1,j+1} = -1.0 \times FPIN \times CMBNM \times ((PS_{i,j+1} + PS_{i+1,j+1}) / 2 - (PS_{i,j} + PS_{i+1,j}) / 2) / DELTYU / FCORSP_{i,j}
\]
\[ G1Y_{i+1,j+1} = FPIN \times CMBNM \times ((PS_{i+1,j} + PS_{i+1,j+1}) / 2 \times DELTYU_j / FCORSP_{i,j} \]

where \( F \) is the Coriolis parameter (1.46 \times 10^{-4}/s), \( RHO \) is the density of air (1.3 km/m^3), \( CMBNM \) is the conversion factor from millibars to N/m^2 (100.0 N/m^2), \( PS \) is the pressure field of atmospheric forcing, \( DELTYU \) is the y (latitude) grid spacing for velocity fields in meters, \( DELTXU \) is the x (longitude) grid spacing for velocity fields in meters, and \( FCORSP \) is the latitude position of the grid element. \( F, RHO, \) and \( CMBNM \) are local constants. \( PS, DELTYU, DELTXU, \) and \( FCORSP \) are inputs to the CSU. The values of \( G1X \) and \( G1Y \) are returned to the calling CSU.

**Data Conversions:**

The pressure must be converted from millibars to N/m^2.

**Logic Flow:**

The RFORCE CSU calls the GEOWIND CSU to calculate the geostrophic winds from the surface pressure fields. The surface air pressure (PS), the array of sine of latitude positions (FCORSP), the x and y grid spacing (DELTXU and DELTYU) are all inputs to the CSU and are used in determining the geostrophic winds.

Before any calculations, the constant values for the Coriolis parameter (F), the density of air (RHO), and the conversion factor from millibars to N/m^2 (CMBNM) are defined:

\[
\begin{align*}
F &= 1.46 \times 10^{-4} \\
RHO &= 1.3 \\
FPIN &= 1.0/(F \times RHO) \\
CMBNM &= 100.0.
\end{align*}
\]

Then the x and y components of the inner geostrophic wind values for each velocity field are defined:

\[
\begin{align*}
G1X(I+1,J+1) &= -1.0 \times FPIN \times CMBNB \times ((PS(I,J+1) + PS(I+1,J+1)) / 2 \times DELTYU/FCORSP(I,J) \\
&\quad - (PS(I,J) + PS(I+1,J)) / 2 \times DELTYU/FCORSP(I,J)
\end{align*}
\]

\[
\begin{align*}
G1X(I+1,J+1) &= FPIN \times CMBNB \times ((PS(I+1,J) + PS(I+1,J+1)) / 2 \times DELTYU/FCORSP(I,J) \\
&\quad - (PS(I,J) + PS(I+1,J+1)) / 2 \times DELTYU/FCORSP(I,J).
\end{align*}
\]

The boundary values are set equal to inside values by first looping over the zonal portion of the grid boxes (I):

\[
\begin{align*}
G1X(I,1) &= G1X(I,2) \\
G1Y(I,1) &= G1Y(I,2) \\
G1X(I,NY2) &= G1X(I,NY1) \\
G1Y(I,NY2) &= G1Y(I,NY1)
\end{align*}
\]

and then looping over the meridian portion of the grid boxes (J):

\[
\begin{align*}
G1X(1,J) &= G1X(2,J) \\
G1Y(1,J) &= G1Y(2,J)
\end{align*}
\]
\[ G1X(NX2,J) = G1X(NX1,J) \]
\[ G1Y(NX2,J) = G1Y(NX1,J). \]

The x and y components of the geostrophic winds are returned to CSU RFORCE through the parameter list.

### 3.3.3 CSC Momentum Balance

The Momentum Balance CSC calculates the terms needed to solve the momentum equation. The CSUs listed below perform the functions required by the CSC Momentum Balance.

- **FORM**: Computes the x and y components of forcing due to the ocean currents plus the ice pressure gradient.
- **RELAX**: Uses the relaxation method to solve the momentum balance equation.
- **QMAX**: Finds the maximum value in an array of values.

The forcing due to the ocean current is estimated in the FORM CSU. The x and y components of forcing are then used by the RELAX CSU to solve the momentum equation through the relaxation method. The CSU QMAX is called by the RELAX CSU to find the largest U and V correction values for each of the grids.

#### 3.3.3.1 CSU FORM

The FORM CSU calculates the forcing terms for the ice momentum balance equation. The “law of the wall” is used to estimate the water/ice drag coefficient and the corresponding turning angle between the top mixed layer and the ice bottom.

#### 3.3.3.1.1 CSU FORM Design Specification/Constraints – There are no known constraints.

#### 3.3.3.1.2 CSU FORM Design

**Input Data Elements:**
- **AREA1**: fraction of grid cell covered by ice
- **ETA**: nonlinear shear viscosity
- **GAIRX**: x component of the geostrophic wind
- **Gairy**: y component of the geostrophic wind
- **GWATX**: x component of ocean current
- **GWATY**: y component of ocean current
- **HEFF**: mean ice thickness per grid cell
- **OUT**: land/sea mask including outflow conditions for thermodynamic variables
- **UICE**: x component of the ice drift
- **VICE**: y component of the ice drift
- **ZETA**: nonlinear bulk viscosity

**/FCORSP/**
- **FCORSP**: array of sine of latitude positions for each gridpoint

**/STEPS/**
- **DELTXU**: x (longitude) grid spacing for velocity fields (m)
- **DELTYY**: y (latitude) grid spacing for velocity fields (m)
Output Data Elements:

AMASS  ice mass per grid area
DRAGA  asymmetric water drag plus the Coriolis parameter
DRAGS  symmetric water drag
ETA    nonlinear shear viscosity
HEFFM  land/sea mask for thermodynamic variables
ZETA   nonlinear bulk viscosity

/FORCE/
FORCEX x component of the forcing due to the ocean currents plus the ice pressure gradient
FORCEY y component of the forcing due to the ocean currents plus the ice pressure gradient

/PRESS/
PRESS  ice strength

Parameters:
stored in file ice.par
The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Local Data Elements:
CDWI   variable drag coefficient between ice and water real
CONSTK von Karman constant (0.41) real
CONSTZ minimum ice roughness (0.01) real
COR(MTM1,JMTM1) array of the Coriolis forcing real
COSWAT cosine of the turning angle for water (0.9063) real
COSWIN cosine of the turning angle for wind (0.9205) real
DAIRN  nonlinear wind drag real
DWATN(MTM1,JMTM1) nonlinear water drag real
ECCEN  ratio of the principal axes of the plastic yield ellipse (2.0) real
FCOR   Coriolis parameter (1.46 × 10^{-4}/s) real
HMIN   minimum ice thickness real
I      index counter integer
II     index counter integer
J      index counter integer
JJ     index counter integer
RHOAIR density of air (1.3 kg/m^3) real
SINWAT sine of the turning angle for water (0.4226) real
SINWIN sine of the turning angle for wind (0.3907) real

Algorithms:
The ice mass per grid cell m is calculated by:

\[ m = \rho_i h \]  

(9)

where \( \rho_i \) is the density of ice 0.91 × 10^3 kg/m^3 and \( h \) is the ice thickness as input to the CSU. The calculated ice mass is output by the CSU and is also used by this CSU to calculate the Coriolis term.
After computing the ice mass, the Coriolis term for each grid cell is calculated using:

\[ \text{COR} = mf \sin(\text{lat}), \] (10)

where \( m \) is the ice mass calculated above, \( f \) is the constant Coriolis parameter value of \( 1.46 \times 10^{-4}/\text{sec} \) and \( \text{lat} \) is the latitude of the grid element in radians.

The equation for air stress is as follows:

\[ \tau_a = \rho_a C_a |\vec{U}_s| \left( \vec{U}_g \cos\phi + \hat{k} \times \vec{U}_g \sin\phi \right), \] (11)

where \( \rho_a \) is the density of air, \( C_a \) is the air drag coefficient, \( \vec{U}_g \) is the geostrophic wind, \( \hat{k} \) is a unit vector normal to the surface, and \( \phi \) is the air turning angle.

The equation for water stress is as follows:

\[ \tau_w = \rho_w C_w |\vec{U}_w - \vec{u}| \left( (\vec{U}_w - \vec{u}) \cos\theta + \hat{k} \times (\vec{U}_w - \vec{u}) \sin\theta \right), \] (12)

where \( \rho_w \) is the density of water, \( C_w \) is the water drag coefficient, \( \vec{U}_w \) is the geostrophic current, \( \vec{u} \) is the ice-drift velocity, \( \theta \) is the water turning angle, and \( \hat{k} \) is a unit vector normal to the surface.

The ice strength is calculated by using the following equation:

\[ P = P^* \; h \exp[-C(1 - A)], \] (13)

where \( P^* \) is the pressure constant \( (2.75 \times 10^4 \; \text{N/m}^2) \), \( C \) is a fixed empirical constant \( (20) \), \( h \) is the ice thickness, and \( A \) is the ice compactness. This relationship allows the ice to become stronger as it thickens.

Data Conversions:

Latitude positions must be converted from degrees to radians before calculations may be performed.

Logic Flow:

The FORM CSU is called by the ICEMDL CSU. It is dedicated to determining the \( x \) and \( y \) components of forcing due to the ocean currents plus the ice pressure gradient. These two equations are stated as:

\[ \text{FORCEX} = X(\text{wind stress}) + X(\text{water drag}) + X(\text{tilt}) + X(\text{ice pressure gradient}) \]
\[ \text{FORCEY} = Y(\text{wind stress}) + Y(\text{water drag}) + Y(\text{tilt}) + Y(\text{ice pressure gradient}). \]

Each \( x \) and \( y \) component (wind stress, water drag, tilt, and ice pressure gradient) is solved and added to the previous value of \( \text{FORCEX} \) and \( \text{FORCEY} \), respectively.

The process of finding the ice pressure gradient and water drag begins by calculating the average ice mass per grid cell (AMASS) and the Coriolis term (COR) for each grid cell. The ice mass per grid cell is calculated by:
AMASS(I,J) = 0.91 \times 10^3 \times 0.25 \times (HEFF(I,J,1) + HEFF(I+1,J,1) + HEFF(I,J+1,1) + HEFF(I+1,J+1,1)),

where HEFF is the mean ice thickness. The Coriolis term for each grid cell is based on the ice mass, Coriolis parameter, and latitude and is computed using:

\[ \text{COR}(I,J) = \text{AMASS}(I,J) \times \text{FCOR} \times \text{FCORSP}(I,J), \]

where AMASS is the ice mass per grid cell, FCOR is the Coriolis parameter constant value of $1.46 \times 10^{-4}$/s and FCORSP is the latitude of the gridpoint in degrees.

Before calculation of the nonlinear water drag, the minimum ice thickness (HMIN) is calculated, values are assigned for von Karman's constant (CONSTK) and the minimum ice roughness (CONSTZ), and value of the variable drag coefficient between ice and water (CDWI) is calculated as follows:

\[ \text{HMIN} = \max (\text{HEFF}(I,J,1), 0.015), \]
\[ \text{CONSTK} = 0.41, \]
\[ \text{CONSTZ} = 0.01, \]
\[ \text{CDWI} = 1.0/(\text{alog}(\text{HMIN}/\text{CONSTZ})/\text{CONSTK}) \times 2, \]
\[ \text{DWATN}(I,J) = 1000 \times \sqrt{((UICE(I,J,1) - GWATX(I,J)) \times 2 + (VICE(I,J,1) - GWATY(I,J)) \times 2) \times \text{CDWI}}, \]

where UICE is the x component of the ice drift, VICE is the y component of the ice drift, GWATX is the x component of the ocean current, and GWATY is the y component of the ocean current. This calculated value of DWATN is set to 0.055 if $<0.055$.

Although the asymmetric (DRAGA) and symmetric (DRAGS) water drags are not used in calculating the forcing fields, the values to be passed to subroutine relax are calculated here using:

\[ \text{DRAGA}(I,J) = \text{DWATN}(I,J) \times \text{SINWAT} + \text{COR}(I,J), \]
\[ \text{DRAGS}(I,J) = \text{DWATN}(I,J) \times \text{COSWAT}, \]

where SINWAT (0.4226) is the sine of the turning angle for water (25°) and COSWAT (0.9063) is the cosine of the turning angle for water (25°).

Processing is continued to solve for the wind stress. The nonlinear wind DAIRN is calculated by:

\[ \text{DAIRN} = \text{RHOAIR} \times 0.0008 \times \sqrt{\text{GAIRX}(I+1,J+1) \times 2 + \text{Gairy}(I+1,J+1) \times 2}, \]

where RHOAIR is the density of air (1.3 kg/m$^3$), 0.0008 is the wind drag coefficient, GAIRX is the x component of the geostrophic wind, and Gairy is the y component of the geostrophic wind.

Once all values necessary for the x and y components due to wind stress are calculated, the initial values for x and y components of forcing due to ocean currents and pressure gradient are computed as follows:

\[ \text{FORCEX}(I,J) = \text{DAIRN}(I,J) \times (\text{COSWIN} \times \text{GAIRX}(I+1,J+1) - \text{SINWIN} \times \text{Gairy}(I+1,J+1)) \]
\[ \text{FORCEY}(I,J) = \text{DAIRN}(I,J) \times (\text{SINWIN} \times \text{GAIRX}(I+1,J+1) + \text{COSWIN} \times \text{Gairy}(I+1,J+1)) \]
where GAIRX is the \( x \) component of the geostrophic wind, GAIRY is the \( y \) component of the geostrophic wind, SINWIN (0.3907) is the sine of the turning angle of air (23\(^\circ\)), and COSWIN (0.9205) is the cosine of the turning angle of air (23\(^\circ\)).

Next, the water stress is added to the existing forcing fields:

\[
\text{FORCEX}(I,J) = \text{FORCEX}(I,J) + \text{DWATN}(I,J) \ast (\text{COSWAT} \ast \text{GWATX}(I,J) \\
- \text{SINWAT} \ast \text{GWATY}(I,J)),
\]

\[
\text{FORCEY}(I,J) = \text{FORCEY}(I,J) + \text{DWATN}(I,J) \ast (\text{SINWAT} \ast \text{GWATX}(I,J) \\
- \text{COSWAT} \ast \text{GWATY}(I,J)),
\]

where GWATX is the \( x \) component of the ocean current, GWATY is the \( y \) component of the ocean current, COSWAT (0.9063) is the cosine of the turning angle of water (25\(^\circ\)), and SINWAT (0.4226) is the sine of the turning angle of water (25\(^\circ\)).

The tilt (or water pressure gradient) in the ice model is computed from the Coriolis parameter and the geostrophic current:

\[
\text{FORCEX}(I,J) = \text{FORCEX}(I,J) - \text{COR}(I,J) \ast \text{GWATY}(I,J),
\]

\[
\text{FORCEY}(I,J) = \text{FORCEY}(I,J) + \text{COR}(I,J) \ast \text{GWATX}(I,J),
\]

where GWATX is the \( x \) component of the ocean current and GWATY is the \( y \) component of the ocean current.

The ice pressure (strength) is calculated using:

\[
\text{PRESS}(I,J) = 2.75 \ast 10^4 \ast \text{HEFF}(I,J,1) \ast \exp(-20.0 \ast (1.0 - \text{AREA1}(I,J,1))),
\]

where HEFF is the mean ice thickness per grid cell and AREA1 is the area of the grid cell covered by ice.

CSU PLAST is called to calculate strain rates (E11, E12, and E22), divergence (DIV), and nonlinear bulk and shear viscosities (ZETA and ETA). After returning from the PLAST CSU, the viscosities and pressure are set equal to 0 at the outflow gridpoints:

\[
\text{ETA}(I,J) = \text{ETA}(I,J) \ast \text{OUT}(I,J),
\]

\[
\text{ZETA}(I,J) = \text{ZETA}(I,J) \ast \text{OUT}(I,J),
\]

\[
\text{PRESS}(I,J) = \text{PRESS}(I,J) \ast \text{OUT}(I,J),
\]

where OUT is the land/sea mask including outflow conditions for thermodynamic variables.

The final step to solving the equation for the \( x \) and \( y \) components of the forcing is to subtract the ice pressure gradient:

\[
\text{FORCEX}(I,J) = \text{FORCEX}(I,J) - (1.0/\text{DELTXU}(J)/2.0) \ast \text{PRESS}(I+1,J) \\
- \text{PRESS}(I,J) + \text{PRESS}(I+1,J+1) - \text{PRESS}(I,J+1))/2.0
\]

\[
\text{FORCEY}(I,J) = \text{FORCEY}(I,J) - (1.0/\text{DELTYU}(2.0) \ast \text{PRESS}(I,J+1) \\
- \text{PRESS}(I,J) + \text{PRESS}(I+1,J+1) - \text{PRESS}(I+1,J))/2.0,
\]
where DELTXU is the x grid spacing in meters for the velocity fields and DELTYU is the y grid spacing in meters for the velocity fields.

This CSU FORM returns the nonlinear shear and bulk viscosities (ETA and ZETA), the ice mass per grid area (AREA1), the symmetric and asymmetric water drags (DRAGS and DRAGA), and the x and y components of forcing (FORCEX and FORCEY) to the ICE CSU upon completion.

**3.3.3.2 CSU RELAX**

The RELAX CSU uses the method of relaxation to solve the momentum equation. This is a successive approximation method for solving systems of equations where the errors from an initial approximation are viewed as constraints to be minimized or relaxed within a toleration limit.

**3.3.3.2.1 CSU RELAX Design Specification/Constraints – There are no known constraints.**

**3.3.3.2.2 CSU RELAX Design**

**Input Data Elements:**

- AMASS: ice mass per grid cell
- DRAGA: asymmetric water drag plus the Coriolis parameter
- DRAGS: symmetric water drag
- ERROR: maximum error allowed in the relation scheme
- ETA: nonlinear shear viscosity
- HEFFM: land/sea mask for thermodynamic variables
- THETA: indicates a backward timestep
- UICE: x component of the ice drift
- UICEC: intermediate x component of the ice drift
- UVM: land/sea mask for the velocity variables
- VICE: y component of the ice drift
- VICEC: intermediate y component of the ice drift
- ZETA: nonlinear bulk viscosity

/CORSP/
FCORSP: array of sine of latitude positions for each gridpoint

/FORCE/
FORCEX: x component of forcing due to the ocean currents plus the ice pressure gradient
FORCEY: y component of forcing due to the ocean currents plus the ice pressure gradient

/STEP/
DELTAT: timestep (s)
DELTAY: y grid spacing (deg)

/STEPSP/
DELTXU: x grid spacing for velocity fields (m)
DELTYU: y grid spacing for velocity fields (m)

**Output Data Elements:**

- UICE: x component of the ice drift
- VICE: y component of the ice drift
Parameters: stored in file ice.par
The ice.par parameters are described in Sec. 3.1.1.2.

stored in file relax.par
N3  360 * 360
N4  359 * 359
MX  359
MY  359
M4X 360/2
M4Y 360/2
M4XY M4X * M4Y
M4XP M4X+1
M4XM M4X–1
M1STOP (359/2–1)M4X + 359/2
M2STOP (359/2–1)M4X + 360/2–1
M3STOP (360/2–2)M4X + 359/2
MSTOP (360/2–1)M4X – 1

Data element design information is provided in Sec. 4.0.

Local Data Elements:

COEFIX(IMTM1,JMTM1) used in determining relaxation coefficients 19, 20, and 21 real
COEFIY(IMTM1,JMTM1) used in determining relaxation coefficients 18, 19, and 21 real
DELIN reciprocal of y grid spacing (m) real
DELIN2 used to define S1TH in spherical coordinates real
DRAG1 used in calculating relaxation coefficients 18, 19, and 21 real
DRAG2 used in calculating relaxation coefficients 19, 20, and 21 real
FX3(M4XY) array for 4-color solver real
FX3A added to FX3 for spherical coordinate system real
FXM(M4XY,4) array for 4-color solver real
FY3(M4XY) array for 4-color solver real
FY3A added to FY3 for spherical coordinate system real
FYM(M4XY,4) array for 4-color solver real
I index counter integer
I4 temporary variable integer
ICOUNT counter for the number of iterations integer
IJ4 index of first dimension for relaxation coefficients integer
J row index counter integer
J4 temporary variable integer
K index of loop over four grids integer
K4 index of third dimension for relaxation coefficients integer
R(M4XY,21,4) relaxation coefficients real
RADIUS degrees per radian (57.29578°) real
RADIUS radius of the earth (6370.0 x 10³ m) real
S1 maximum error for u and v real
S11 largest value of the u velocity for grid 1 real
S12 largest value of the u velocity for grid 2 real
S13 largest value of the u velocity for grid 3 real
S14 \text{largest value of the u velocity for grid 4} \quad \text{real}

S1TH \text{used in defining relaxation} \quad \text{real}

\text{coefficients 5, 6, 7, 8, 9, 10, 11, 12, and 13}

S21 \text{largest value of the v velocity for grid 1} \quad \text{real}

S22 \text{largest value of the v velocity for grid 2} \quad \text{real}

S23 \text{largest value of the v velocity for grid 3} \quad \text{real}

S24 \text{largest value of the v velocity for grid 4} \quad \text{real}

S2TH \text{used in defining relaxation coefficients 1, 4, 14, and 17} \quad \text{real}

U4(M4XY,4) \text{x component of the ice drift} \quad \text{real}

U4C(M4XY) \text{u velocity correction} \quad \text{real}

V4(M4XY,4) \text{y component of the ice drift} \quad \text{real}

V4C(M4XY) \text{v velocity correction} \quad \text{real}

WFA \text{relaxation factor (1.5)} \quad \text{real}

\textbf{Algorithms:}

Given the present ice rheology, there are 17 independent coefficients necessary for the relaxation steps. These coefficients are calculated by using the following equations:

\begin{align*}
R_1 &= (1/2/\partial y/\partial y)(\eta_{i,j} + \eta_{i+1,j}) + (m_{i,j} \, \nu c_{i,j}/\partial y/2) \\
R_2 &= (1/2/\partial x/\partial x)(\eta_{i,j} + \eta_{i+1,j} + \zeta_{i,j} + \zeta_{i+1,j}) + m_{i,j} \, \nu c_{i,j}/\partial x/2 \\
R_3 &= (1/2/\partial y/\partial x)(\eta_{i+1,j} + \eta_{i+1,j} + \zeta_{i+1,j} + \zeta_{i+1,j}) - m_{i,j} \, \nu c_{i,j}/\partial x/2 \\
R_4 &= (1/2/\partial y/\partial y)(\eta_{i+1,j} + \eta_{i+1,j} + \eta_{i+1,j}) - (m_{i,j} \, \nu c_{i,j}/\partial y/2) \\
R_5 &= (1/4/\partial x/\partial y)(\zeta_{i,j}) \\
R_6 &= (1/4/\partial x/\partial y)[2(\eta_{i,j} - \eta_{i+1,j}) - (\eta_{i,j} + \zeta_{i,j})] \\
R_7 &= (1/4/\partial x/\partial y)(- \zeta_{i+1,j}) \\
R_8 &= (1/4/\partial x/\partial y)[2(\eta_{i,j} + \eta_{i+1,j} - (\eta_{i,j} + \zeta_{i,j})] \\
R_9 &= (1/4/\partial x/\partial y)(\zeta_{i+1,j} + \zeta_{i,j} - \zeta_{i,j} - \zeta_{i+1,j}) \\
R_{10} &= (1/4/\partial x/\partial y)[2(\eta_{i+1,j} + \eta_{i+1,j} - (\zeta_{i,j} + \zeta_{i,j} + \zeta_{i+1,j})] \\
R_{11} &= (1/4/\partial x/\partial y)(- \zeta_{i,j} + 1) \\
R_{12} &= (1/4/\partial x/\partial y)[2(\eta_{i,j} + \eta_{i+1,j} + \eta_{i+1,j} - (\zeta_{i,j} + 1 - \zeta_{i+1,j} + 1)] \\
R_{13} &= (1/4/\partial x/\partial y)(\zeta_{i,j} + 1) \\
R_{14} &= (1/2/\partial y/\partial y)(\eta_{i,j} + \eta_{i+1,j} + \zeta_{i,j} + \zeta_{i+1,j}) + m_{i,j} \, \nu c_{i,j}/\partial y/2 \\
R_{15} &= (1/2/\partial x/\partial y)(\eta_{i,j} + \eta_{i+1,j}) + (m_{i,j} \, \nu c_{i,j}/\partial x/2) \\
R_{16} &= (1/2/\partial x/\partial y)(\eta_{i+1,j} + \eta_{i+1,j} + \eta_{i+1,j} - m_{i,j} \, \nu c_{i,j}/\partial x/2 \\
R_{17} &= (1/2/\partial y/\partial y)(\eta_{i+1,j} + \eta_{i+1,j} + \zeta_{i+1,j} + \zeta_{i+1,j} + \zeta_{i+1,j} + 1) - m_{i,j} \, \nu c_{i,j}/\partial y/2. \quad (14)
\end{align*}
Here, $dy$ is the y grid spacing in meters, $dx$ is the x grid spacing in meters, $\eta$ is the nonlinear shear viscosity, $\zeta$ is the nonlinear bulk viscosity, $vc$ is the intermediate y component of the ice drift for use in the semi-implicit timestepping, $uc$ is the intermediate x component of the ice drift for use in the semi-implicit timestepping, and $m$ is the ice mass per grid area.

To perform the relaxation, the 17 independent coefficients are used to calculate the following equations:

$$RHS_x = FXM_{ij} + \begin{vmatrix} 0 & R_{4} & 0 \\ R_{2} & 0 & R_{3} \\ 0 & R_{1} & 0 \end{vmatrix} U_{ij} + \begin{vmatrix} R_{11} & R_{12} & R_{13} \\ R_{8} & R_{9} & R_{10} \\ R_{5} & R_{6} & R_{7} \end{vmatrix} V_{ij}$$

$$RHS_y = FYM_{ij} + \begin{vmatrix} R_{11} & -R_{12} & R_{13} \\ -R_{8} & R_{9} & -R_{10} \\ R_{5} & -R_{6} & R_{7} \end{vmatrix} U_{ij} + \begin{vmatrix} 0 & R_{17} & 0 \\ R_{15} & 0 & R_{16} \\ 0 & R_{14} & 0 \end{vmatrix} V_{ij} \quad (15)$$

Data Conversions:
Latitude positions are converted from degrees to radians.

Logic Flow:
The RELAX CSU is called by the ICEMDL CSU.

Upon entering this CSU, the values of R, FXM, FYM, U4, V4, U4C, V4C, FX3, and FY3 are initialized to 0. The value of ICOUNT is set to 0 and the relaxation factor WFA is set to 1.5. DELIN is set to the reciprocal of the y (latitude) grid spacing:

$$DELIN = 1.0/DELYTU$$
$$DELIN2 = DELIN \times DELIN/2.0$$

Then the x and y components of the ice drift at the second time level are set equal to the one at the first:

$$UICE(I,J,2) = UICE(I,J,1)$$
$$VICE(I,J,2) = VICE(I,J,1)$$

The ice drift at the boundary points are set equal to 0 by multiplying the ice drift by the land/sea mask:

$$UICE(I,J,1) = UICE(I,J,3) \times UVM(I,J)$$
$$VICE(I,J,1) = VICE(I,J,3) \times UVM(I,J)$$

COEFFIX and COEFFFY are defined in spherical coordinates as

$$COEFFIX(I,J) = 1.0/AMASS(I,J)/DELTAT + \Theta \times (DRAGS(I,J) + ETA(I,J) + ETA(I+1,J))$$
+ ETA(I,J+1) + ETA(I+1,J+1))/2.0 DELTYU ** 2
+ (ETA(I,J) + ETA(I+1,J) + ETA(I,J+1) + ETA(I+1,J+1) + ZETA(I,J)
+ ZETA(I+1,J) + ZETA(I,J+1) + ZETA(I+1,J+1))/2.0/DELTXU(J) ** 2)

COEFIGY(I,J) = 1.0/(AMASS(I,J)/DELTAT
+ THETA * (DRAGS(I,J) + ETA(I,J) + ETA(I+1,J) + ETA(I,J+1)
+ ETA(I+1,J+1))/2.0/DELTXU(J) ** 2
+ (ETA(I,J) + ETA(I+1,J) + ETA(I,J+1)
+ ETA(I+1,J+1) + ZETA(I,J) + ZETA(I+1,J) + ZETA(I,J+1)
+ ZETA(I+1,J+1))/2.0/DELTYU ** 2)),

where THETA = 1 for the Euler backward iteration.

The values of FY3A and FX3A are 0

FY3A = - tan(DELTAY * (J-(MIDY-0.5) - 0.5)/RADIANT/RADIUS) * 2.0
* (ETA(I,J) + ETA(I-1,J) + ETA(I,J-1) + ETA(I-1,J-1))/4.0)
* ((VICE(I,J+1,1) - VICE(I,J-1,1))/2.0/DELTYU) * UVM(I,J)

FX3A = - tan(DELTAY * (J-(MIDY-0.5) - 0.5)/RADIANT/RADIUS)
* (ETA(I,J) + ETA(I-1,J) + ETA(I,J-1) + ETA(I-1,J-1))/4.0)
* ((VICE(I+1,J,1) - VICE(I-1,J,1))/2.0/DELTXU(J)
+ (VICE(I,J+1,1) UICE(I,J+1,1))/2.0/DELTYU) * UVM(I,J)

FXM(IJ4,K4) = UICE(I,J,2) * AMASS(I,J)/DELTAT + FORCEX(I,J) + FX3A

FYM(IJ4,K4) = VICE(I,J,2) * AMASS(I,J)/DELTAT + FORCEY(I,J) + FY3A,

where DELTAY is the y grid spacing in degrees, MIDY is equal to 160.0, RADIANT is the conversion factor from degrees to radians, RADIUS is the radius of the earth in meters, ETA is the nonlinear shear viscosity, VICE is the y component of the ice drift, DELTYU is the y grid spacing in meters, UVM is the land/sea mask for the velocity fields, DELTXU is the x grid spacing in meters, and UICE is the x component of the ice drift.

At this point, the relaxation coefficients are explicitly calculated. S1TH, S2TH, and SDL are then defined in spherical coordinates:

S1TH = 0.5 * DELIN2 * THETA
S2TH = 2.0 * S1TH
SDL = 0.5 * THETA * DELIN

The relaxation coefficients are defined for four grids (K4):

J4 = IJ4/M4X + 1
I4 = IJ4 - (J4 - 1) * M4X
J = 2 * J4 - (4 - K4)/2.
If the value of $J$ is greater than 1 or less than the number of rows for the velocity fields, then the value of $I$ is calculated as:

$$I = 2 \ast I4 - \text{mod}(K4,2).$$

If the value of $I$ is greater than 1 or less than the number of columns for the velocity fields, then the relaxation coefficients are computed by:

$$R(IJ4,1,K4) = S2TH \ast (ETA(I,J) + ETA(I+1,J)) + SDL \ast AMASS(I,J) \ast VICEC(I,J)$$

$$R(IJ4,2,K4) = (ETA(I,J) + ETA(I,J) + ZETA(I,J+1) + ZETA(I,J))$$
$$\ast THETA(2.0 \ast DELTXU(J) \ast DELTXU(J)) + AMASS(I,J)$$
$$\ast UICEC(I,J) \ast THETA(2.0 \ast DELTXU(J))$$

$$R(IJ4,3,K4) = (ETA(I+1,J+1) + ETA(I+1,J) + ZETA(I+1,J+1) + ZETA(I+1,J))$$
$$\ast THETA(2.0 \ast DELTXU(J) \ast DELTXU(J)) - AMASS(I,J)$$
$$\ast UICEC(I,J) \ast THETA(2.0 \ast DELTXU(J))$$

$$R(IJ4,4,K4) = S2TH \ast (ETA(I+1,J+1) + ETA(I,J+1)) - AMASS(I,J) \ast VICEC(I,J) \ast SDL.$$

The $S1TH$ term is redefined in spherical coordinates as follows:

$$S1TH = THETA \ast DELIN/4.0/DELTXU(J)$$

$$R(IJ4,5,K4) = S1TH \ast ZETA(I,J)$$

$$R(IJ4,6,K4) = - S1TH \ast (2.0 \ast (ETA(I,J) - ETA(I+1,J)) + ZETA(I+1,J) - ZETA(I,J))$$

$$R(IJ4,7,K4) = - S1TH \ast ZETA(I+1,J)$$

$$R(IJ4,8,K4) = - S1TH \ast (2.0 \ast (ETA(I,J+1) - ETA(I,J)) + ZETA(I,J) - ZETA(I,J+1))$$

$$R(IJ4,9,K4) = S1TH \ast (ZETA(I+1,J) + ZETA(I,J+1) - ZETA(I,J) - ZETA(I+1,J+1))$$

$$R(IJ4,10,K4) = S1TH \ast (2.0 \ast (ETA(I+1,J+1) - ETA(I+1,J)) + ZETA(I+1,J)$$
$$\ast ZETA(I+1,J+1))$$

$$R(IJ4,11,K4) = - S1TH \ast ZETA(I,J+1)$$

$$R(IJ4,12,K4) = S1TH \ast (2.0 \ast (ETA(I,J+1) - ETA(I+1,J+1)) + (ZETA(I+1,J+1)$$
$$\ast ZETA(I,J+1)))$$

$$R(IJ4,13,K4) = S1TH \ast ZETA(I+1,J+1)$$

$$R(IJ4,14,K4) = S2TH \ast (ETA(I,J) + ETA(I+1,J) + ZETA(I,J) + ZETA(I+1,J))$$
$$+ SDL \ast AMASS(I,J) \ast VICEC(I,J)$$

$$R(IJ4,15,K4) = (ETA(I,J) + ETA(I,J+1)) \ast THETA(2.0 \ast DELTXU(J)$$
$$\ast DELTXU(J)) + AMASS(I,J) \ast UICEC(I,J) \ast THETA(2.0$$
$$\ast DELTXU(J))$$

$$R(IJ4,16,K4) = (ETA(I+1,J+1) + ETA(I+1,J)) \ast THETA(2.0 \ast DELTXU(J)$$
$$\ast DELTXU(J)) - AMASS(I,J) \ast UICEC(I,J) \ast THETA(2.0$$
$$\ast DELTXU(J))$$

$$R(IJ4,17,K4) = S2TH \ast (ETA(I+1,J+1) + ETA(I,J+1) + ZETA(I+1,J+1)$$
$$+ ZETA(I,J+1)) - AMASS(I,J) \ast VICEC(I,J) \ast SDL$$

$$DRAG1 = (DRAGA(I,J) + R(IJ4,9,K4)) \ast THETA$$
\[ \text{DRAG2} = (\text{DRAGA}(I,J) - R(IJ4,9,K4)) \times \text{THETA} \]

\[ R(IJ4,18,K4) = \text{THETA} \times \text{COEFIY}(I,J) \times \text{DRAG1} \]

\[ R(IJ4,19,K4) = \text{COEFIX}(I,J) \times \text{UVM}(I,J)/\]

\[ (1.0+\text{DRAG1} \times \text{DRAG2} \times \text{COEFIX}(I,J) \times \text{COEFIY}(I,J)) \]

\[ R(IJ4,20,K4) = \text{THETA} \times \text{COEFIX}(I,J) \times \text{DRAG2} \]

\[ R(IJ4,21,K4) = \text{COEFIY}(I,J) \times \text{UVM}(I,J)/\]

\[ (1.0+\text{DRAG1} \times \text{DRAG2} \times \text{COEFIX}(I,J) \times \text{COEFIY}(I,J)). \]

Once all of the relaxation coefficients are calculated, the relaxation can be done directly. First, UICE and VICE are loaded into U4 and V4, respectively:

\[ U4(IJ4,K4) = \text{UICE}(I,J,1) \]

\[ V4(IJ4,K4) = \text{VICE}(I,J,1) \]

The iterative process begins here by performing a relaxation on grid 1:

\[ \text{FX3}(I) = \text{FXM}(I,1) + U4(I-1,2) \times R(I,2,1) + U4(I,2) \times R(I,3,1) + \]

\[ + U4(I-M4X,3) \times R(I,1,1) + U4(I,3) \times R(I,4,1) + V4(I-1,2) \times R(I,8,1) \]

\[ + V4(I,2) \times R(I,10,1) + V4(I-M4X,3) \times R(I,6,1) + V4(I,3) \times R(I,12,1) \]

\[ + V4(I-M4XP,4) \times R(I,5,1) + V4(I,4) \times R(I,13,1) + V4(I-1,4) \times R(I,11,1) \]

\[ + V4(I-M4X,4) \times R(I,7,1), \]

\[ \text{FY3}(I) = \text{FYM}(I,1) + V4(I-1,2) \times R(I,15,1) + V4(I,2) \times R(I,16,1) \]

\[ + V4(I-M4X,3) \times R(I,14,1) + V4(I,3) \times R(I,17,1) - U4(I-1,2) \times R(I,8,1) \]

\[ - U4(I,2) \times R(I,10,1) - U4(I-M4X,3) \times R(I,6,1) - U4(I,3) \times R(I,12,1) \]

\[ + U4(I-M4XP,4) \times R(I,5,1) + U4(I,4) \times R(I,13,1) + U4(I-1,4) \times R(I,11,1) \]

\[ + U4(I-M4X,4) \times R(I,7,1), \]

where \( I = \text{M4XP} \) to \( \text{M1STOP} \).

The arrays for the velocity corrections are initialized to 0:

\[ U4C(I) = 0, \]

\[ V4C(I) = 0, \]

where \( I = 1 \) to \( \text{M4X} \).

\[ U4C(I) = 0, \]

\[ V4C(I) = 0, \]

where \( I = \text{M1STOP} \) to \( \text{M4XY} \).

The velocity corrections are then calculated:

\[ U4C(I) = ((\text{FX3}(I)+\text{FY3}(I) \times R(I,18,1)) \times R(I,19,1) - U4(I,1)) \times \text{WFA} \]

\[ V4C(I) = ((\text{FY3}(I) - \text{FX3}(I) \times R(I,20,1)) \times R(I,21,1) - V4(I,1)) \times \text{WFA}, \]

where \( I = \text{M4XP} \) to \( \text{M1STOP} \).
The velocity corrections are then added to the U4 and V4 values for each \( I = \text{M4XP} \) to M1STOP:

\[
U4(I,1) = U4(I,1) + U4C(I) \\
V4(I,1) = V4(I,1) + V4C(I)
\]

The CSU QMAX is called to find the largest correction for both U4C and V4C. The values are returned as S11 and S21, where S11 is the largest correction of the U4C array and S21 is the largest correction of the S21 array.

Once the relaxation on grid 1 is complete, the relaxation is performed on grid 2:

\[
FX3(I) = FXM(I,2) + U4(I,1) \times R(I,2,2) + U4(I+1,1) \times R(I,3,2) \\
+ U4(I-M4X,4) \times R(I,1,2) + U4(I,4) \times R(I,4,2) + V4(I,1) \times R(I,8,2) \\
+ V4(I+1,1) \times R(I,10,2) + V4(I-M4X,4) \times R(I,6,2) + V4(I,4) \times R(I,12,2) \\
+ V4(I-M4XP3,3) \times R(I,5,2) + V4(I+1,3) \times R(I,13,2) + V4(I,3) \times R(I,11,2) \\
+ V4(I-M4XM3,3) \times R(I,7,2)
\]

\[
FY3(I) = FYM(I,2) + V4(I,1) \times R(I,15,2) + V4(I+1,1) \times R(I,16,2) \\
+ V4(I-M4X,4) \times R(I,14,2) + V4(I,4) \times R(I,17,2) - U4(I,1) \times R(I,8,2) \\
- U4(I+1,1) \times R(I,10,2) - U4(I-M4X,4) \times R(I,6,2) - U4(I,4) \times R(I,12,2) \\
+ U4(I-M4X,3) \times R(I,5,2) + U4(I+1,3) \times R(I,13,2) + U4(I,3) \times R(I,11,2) \\
+ U4(I-M4XM3,3) \times R(I,7,2)
\]

where \( I = \text{M4X} \) to M2STOP.

The U4C and V4C array are cleaned up by:

\[
U4C(M1STOP) = 0, \\
V4C(M1STOP) = 0.
\]

The velocity corrections for grid 2 are then calculated:

\[
U4C(I) = ((FX3(I)+FY3(I) \times R(I,18,2)) \times R(I,19,2) - U4(I,2)) \times WFA, \\
V4C(I) = ((FY3(I) - FX3(I) \times R(I,20,2)) \times R(I,21,2) - V4(I,2)) \times WFA,
\]

where \( I = \text{M4X} \) to M2STOP.

The velocity corrections for grid 2 are then added to the U4 and V4 values for each \( I = \text{M4X} \) to M2STOP:

\[
U4(I,2) = U4(I,2) + U4C(I) \\
V4(I,2) = V4(I,2) + V4C(I)
\]

The CSU QMAX is called to find the largest correction for both U4C and V4C. The values are returned as S12 and S22.

Once the relaxation on grid 2 is complete, the relaxation is performed on grid 3:

\[
FX3(I) = FXM(I,3) + U4(I-1,4) \times R(I,2,3) + U4(I,4) \times R(I,3,3) \\
+ U4(I,1) \times R(I,1,3) + U4(I-M4X,1) \times R(I,4,3) + V4(I-1,4) \times R(I,8,3) \\
+ V4(I,4) \times R(I,10,3) + V4(I,1) \times R(I,6,3) + V4(I+M4X,1) \times R(I,12,3) \\
+ V4(I-1,2) \times R(I,5,3) + V4(I+M4X,2) \times R(I,13,3) \\
+ V4(I+M4XM2,2) \times R(I,11,3) + V4(I,2) \times R(I,7,3)
\]
\[ \text{FY3}(I) = \text{FYM}(I, 3) + V4(I-1, 4) \times R(I, 15, 3) + V4(I, 4) \times R(I, 16, 3) + V4(I, 1) \times R(I, 14, 3) + V4(I+M4X, 1) \times R(I, 17, 3) - U4(I-1, 4) \times R(I, 8, 3) - U4(I, 4) \times R(I, 10, 3) - U4(I, 1) \times R(I, 6, 3) - U4(I+M4X, 1) \times R(I, 12, 3) + U4(I-1, 2) \times R(I, 5, 3) + U4(I+M4X, 2) \times R(I, 13, 3) + U4(I+M4XM, 2) \times R(I, 11, 3) + U4(I, 2) \times R(I, 7, 3), \]

where \( I = M4X \) to M3STOP.

The U4C and V4C array are cleaned up by:

\[ \text{U4C}(I) = 0, \]
\[ \text{V4C}(I) = 0, \]

where \( I = M3STOP \) to M2STOP.

The velocity corrections for grid 3 are then calculated:

\[ \text{U4C}(I) = ((\text{FX3}(I) + \text{FY3}(I) \times R(I, 18, 3)) \times R(I, 19, 3) - U4(I, 3)) \times WFA, \]
\[ \text{V4C}(I) = ((\text{FY3}(I) - \text{FX3}(I) \times R(I, 20, 3)) \times R(I, 21, 3) - V4(I, 3)) \times WFA, \]

where \( I = 2 \) to M3STOP.

The velocity corrections for grid 3 are then added to the U4 and V4 values for each \( I = 2 \) to M3STOP:

\[ \text{U4}(I, 3) = \text{U4}(I, 3) + \text{U4C}(I) \]
\[ \text{V4}(I, 3) = \text{V4}(I, 3) + \text{V4C}(I) \]

The CSU QMAX is called to find the largest correction for both U4C and V4C. The values are returned as S13 and S23.

When the relaxation on grid 3 is complete, the relaxation is performed on grid 4:

\[ \text{FX3}(I) = \text{FXM}(I, 4) + U4(I, 3) \times R(I, 2, 4) + U4(I+1, 3) \times R(I, 3, 4) + U4(I, 2) \times R(I, 1, 4) + U4(I+M4X, 2) \times R(I, 4, 4) + V4(I, 3) \times R(I, 8, 4) + V4(I+1, 3) \times R(I, 10, 4) + V4(I, 2) \times R(I, 6, 4) + V4(I+M4X, 2) \times R(I, 12, 4) + V4(I, 1) \times R(I, 5, 4) + V4(I+M4XP, 1) \times R(I, 13, 4) + V4(I+M4X, 1) \times R(I, 11, 4) + V4(I+1, 1) \times R(I, 7, 4) \]
\[ \text{FY3}(I) = \text{FYM}(I, 4) + V4(I, 3) \times R(I, 15, 4) + V4(I+1, 3) \times R(I, 16, 4) + V4(I, 2) \times R(I, 14, 4) + V4(I+M4X, 2) \times R(I, 17, 4) - U4(I, 3) \times R(I, 8, 4) - U4(I+1, 3) \times R(I, 10, 4) - U4(I, 2) \times R(I, 6, 4) - U4(I+M4X, 2) \times R(I, 12, 4) + U4(I, 1) \times R(I, 5, 4) + U4(I+M4XP, 1) \times R(I, 13, 4) + U4(I+M4X, 1) \times R(I, 11, 4) + U4(I+1, 1) \times R(I, 7, 4), \]

where \( I = 1 \) to M4STOP.

The U4C and V4C array are cleaned up by:

\[ \text{U4C}(\text{M3STOP}) = 0, \]
\[ \text{V4C}(\text{M3STOP}) = 0. \]
The velocity corrections for grid 4 are then calculated:

\[ U_{4C}(I) = ((FX3(I)+FY3(I) \times R(I,18,4)) \times R(I,19,4) - U_4(I,4)) \times \text{WFA}, \]

\[ V_{4C}(I) = ((FY3(I) - FX3(I) \times R(I,20,4)) \times R(I,21,4) - V_4(I,4)) \times \text{WFA}, \]

where \( I = 1 \) to M4STOP.

The velocity corrections for grid 4 are then added to the U4 and V4 values for each \( I = 1 \) to M4STOP:

\[ U_4(I,3) = U_4(I,3) + U_{4C}(I) \]
\[ V_4(I,3) = V_4(I,3) + V_{4C}(I) \]

The CSU QMAX is called to find the largest correction for both U4C and V4C. The values are returned as S14 and S24.

The iteration loop counter ICOUNT is incremented. If the counter is \( \leq 2000 \), then the maximum velocity correction from all four grids is determined. If this value is less than or equal to the maximum error allowed in the relation scheme (ERROR), then the values of UICE and VICE are reloaded; otherwise, the iterative process continues by relaxing on all four grids again. If the number of iterations exceeds 2000, then the convergence fails and a message is sent to standard output. The values of UICE and VICE are reloaded as

\[ \text{DO } K4 = 1, 4 \]
\[ \text{DO } IJ4 = 1, \text{ M4XY} \]
\[ J4 = IJ4/M4X + 1 \]
\[ J4 = IJ4 - (J4 - 1) \times M4X \]
\[ J = 2 \times J4 - (4 - K4)/2 \]
\[ \text{IF (.NOT. (J .LE. 1 .OR. J .GE. JMTM1)) THEN} \]
\[ I = 2 \times J4 - \text{MOD(K4,2)} \]
\[ \text{IF (.NOT. (I .LE. 1 .OR. I .GE. IMTM1)) THEN} \]
\[ \text{UICE}(I,J,1) = U_4(IJ4,K4) \]
\[ \text{VICE}(I,J,1) = V_4(IJ4,K4) \]

\[ \text{END IF} \]
\[ \text{END IF} \]
\[ \text{CONTINUE} \]
\[ \text{CONTINUE.} \]

The number of iterations and the maximum error for U and V are written to standard output.

### 3.3.3.3 CSU QMAX

The CSU QMAX finds the maximum value of an array.

#### 3.3.3.3.1 CSU QMAX Design Specification/Constraints – There are no known constraints.

#### 3.3.3.3.2 CSU QMAX Design

**Input Data Elements:**

<table>
<thead>
<tr>
<th>U</th>
<th>Input array #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>Input array #2</td>
</tr>
</tbody>
</table>
Output Data Elements:
S11  Maximum value in array #1
S21  Maximum value in array #2
U    Input array #1
V    Input array #2

Parameters:
stored in file ice.par
The ice.par parameters are described in Sec. 3.1.1.2.

stored in file qmax.par
M4X   IMT/2; = 180
M4Y   JMT/2; = 180
M4XY  M4X * M4Y; = 32,400

Data element design information is provided in Sec. 4.0.

Local Data Elements:
N     index counter

Logic Flow:
This CSU is called by the RELAX CSU.
The initial maximum values are set to the first element of each array:
U(N) = abs(U(N))
V(N) = abs(V(N)).
The initial maximum values are set to the first element of each array:
S11 = U(1)
S21 = V(1).
The maximum value of each array is then found and control is returned the RELAX CSU:
S11 = max(S11,U(N))
S21 = max(S21,V(N)).

3.3.4 CSC Ice Rheology
The Ice Rheology CSC calculates the ice stress that is directly related to the ice strength and strain rates.

3.3.4.1 CSU PLAST Computer Software Unit
The purpose of the PLAST CSU is to calculate strain rates, divergence, and viscosities based on plastic flow specified by an elliptic yield curve.

3.3.4.1.1 CSU PLAST Design Specification/Constraints – There are no known constraints.
3.3.4.1.2 CSU PLAST Design

Input Data Elements:
- ECCEN: ratio of the principal axes of the plastic yield ellipse
- ETA: nonlinear shear viscosity
- HEFFM: thermodynamic land/sea mask
- PRESS: ice strength
- UIICE: x component of the ice drift
- VICE: y component of the ice drift

/STEP/
- DELTAX: x (longitude) grid spacing (deg)
- DELTAY: y (latitude) grid spacing (deg)

/STEPSP/
- DELTXA: x (longitude) grid spacing for thermodynamic fields (m)
- DELTYA: y (latitude) grid spacing for thermodynamic fields (m)

Output Data Elements:
- ZETA: nonlinear bulk viscosity

Parameters:
stored in file ice.par
The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Local Data Elements:
- DELT: used in calculating ETA and ZETA
- DELT1: square root of DELT
- E11: xx strain component
- E12: xy strain component
- E22: yy strain component
- ECM2: reciprocal of ECCEN²
- GMIN: minimum value of DELT1 (1.0 × 10⁻²⁰)
- I: index counter
- J: index counter
- ZMAX: maximum allowable bulk viscosity value
- ZMIN: minimum allowable bulk viscosity value (4.0 × 10⁸)

Algorithms:
Strain rates, averaged over the grid cell, are estimated for use in solving the momentum equations. The xx \( \dot{\varepsilon}_{11} \), yy \( \dot{\varepsilon}_{22} \), and xy \( \dot{\varepsilon}_{12} \) strain components are obtained by using the following equations:

\[
\begin{align*}
\left( \dot{\varepsilon}_{11} \right)_{i+1/2, j+1/2} &= \frac{1}{2h} \left[ u_{i+1,j+1} + u_{i+1,j-1} + u_{i,j+1} - u_{i,j-1} \right], \\
\left( \dot{\varepsilon}_{22} \right)_{i+1/2, j+1/2} &= \frac{1}{2h} \left[ v_{i+1,j+1} + v_{i,j+1} + v_{i+1,j} - v_{i,j} \right], \\
\left( \dot{\varepsilon}_{12} \right)_{i+1/2, j+1/2} &= \frac{1}{4h} \left[ u_{i+1,j+1} + u_{i+1,j-1} + u_{i,j+1} - u_{i,j-1} \right] \\
&\quad + \frac{1}{4h} \left[ v_{i+1,j+1} + v_{i+1,j} - v_{i,j+1} - v_{i,j} \right],
\end{align*}
\] (16)
where \( h \) is either the \( x \) or \( y \) grid spacing, \( u \) is the \( x \) component of the ice drift, and \( v \) is the \( y \) component of the ice drift.

The nonlinear bulk (\( \zeta \)) and shear (\( \eta \)) viscosities are calculated using the strain rates from above, the pressure term (\( P/2 \)), and the ratio of the principal axes of the ellipse (\( e \)). These terms are stated as

\[
\Delta = \left[ \left( \dot{\varepsilon}_{11}^2 + \dot{\varepsilon}_{22}^2 \right) \left( 1 + 1/e^2 \right) + 4e^{-2} \dot{\varepsilon}_{12}^2 + 2e^{-2} \dot{\varepsilon}_{11}^2 \dot{\varepsilon}_{22} \left( 1 - 1/e^2 \right) \right]^{1/2},
\]

\[
\zeta = P/2\Delta, \\
\eta = \zeta/e^2.
\]

(17)

**Logic Flow:**

The PLAST CSU is called by the FORM CSU to solve the equations for the strain rates, viscosities, and divergence.

First, the values necessary for solving the viscosity equations are calculated by:

\[
ECM2 = 1.0/(ECCEN^2),
\]

where ECCEN is the ratio of the principal axes of the plastic yield ellipse. Then the constant values for the minimum allowable bulk viscosity (ZMIN) and the minimum allowable value of DELT1 are set:

\[
ZMIN = 4.0 \times 10^8 \\
GMIN = 1.0 \times 10^{-20}
\]

Next, the xx (E11), yy (E22), and xy (E12) strain rates for each \( x \) (I) and \( y \) (J) component of the velocity fields are evaluated by averaging the \( x \) and \( y \) components of the ice drift over the grid cell. The strain rates are used in calculating the nonlinear viscosities:

\[
E11(I,J) = (0.5/DELTAX) \\
\times (UICE(I,J,1) + UICE(I,J-1,1) - UICE(I-1,J,1) - UICE(I-1,J-1,1)),
\]

\[
E22(I,J) = (0.5/DELTAY) \\
\times (VICE(I,J,1) + VICE(I,J-1,1) - VICE(I-1,J,1) - VICE(I-1,J-1,1)),
\]

\[
E12(I,J) = (0.25/DELTAX) \\
\times (UICE(I,J,1) + UICE(I,J-1,1) - UICE(I,J-1,1) - UICE(I-1,J,1)), \\
+ (0.25/DELTAX) \\
\times (VICE(I,J,1) + VICE(I,J-1,1) - VICE(I,J-1,1) - VICE(I-1,J,1)),
\]

where DELTAX is the \( x \) (longitude) grid spacing in degrees, DELTAY is the \( y \) (latitude) grid spacing in degrees, UICE is the \( x \) component of the ice drift, and VICE is the \( y \) component of the ice drift. \( E11, E22, \) and \( E12 \) are then redefined in the spherical coordinate system by:

\[
E11(I,J) = E11(I,J) \times DELTAX/DELTXA(J),
\]

\[
E22(I,J) = E22(I,J) \times DELTAY/DELTYA,
\]
\[ E12(I,J) = (1.0/2.0/DELYA) \]
\[ * (UICE(I,J,1) + UICE(I-1,J,1) - UICE(I,J-1,1) - UICE(I-1,J-1,1))/2.0 \]
\[ + (1.0/2.0/DELYA) \]
\[ * (VICE(I,J,1) + VICE(I-1,J,1) - VICE(I,J-1,1) - VICE(I-1,J-1,1))/2.0, \]

where DELYA and DELXY are the x and y grid spacings, respectively, for thermodynamic fields.

After setting up the strain rates, the nonlinear bulk viscosity (ZETA) for each velocity field is calculated:

\[ \text{DELT} = (E11(I,J) ** 2 + E22(I,J) ** 2) * (1.0 + ECM2) + 4.0 * ECM2 * E12(I,J) ** 2 \]
\[ + 2.0 * E11(I,J) * E22(I,J) * (1.0 - ECM2), \]
\[ \text{DELT1} = \sqrt{\text{DELT}}, \]
\[ \text{DELT1} = \max(GMIN, \text{DELT1}), \]
\[ \text{ZETA}(I,J) = 0.5 * \text{PRESS}(I,J)/\text{DELT1}, \]

where PRESS is the ice strength. Note that ZETA cannot be less than \( 1.0 \times 10^{-20} \).

The minimum and maximum values of ZETA are set to:

\[ \text{ZMIN} = 4.0 \times 10^8, \]
\[ \text{ZMAX} = ((5.0 \times 10^{12})/(2.0 \times 10^4)) * \text{PRESS}(I,J). \]

The value of ZMAX is based on the pressure for each thermodynamic field. The value of ZETA must be within these minimum and maximum values.

The nonlinear shear viscosity (ETA) can then be calculated for each thermodynamic field:

\[ \text{ETA}(I,J) = ECM2 * \text{ZETA}(I,J). \]

The strain rates are zeroed out at the land/sea boundary points for the thermodynamic fields:

\[ E11(I,J) = E11(I,J) * \text{HEFFM}(I,J) \]
\[ E22(I,J) = E22(I,J) * \text{HEFFM}(I,J) \]
\[ E12(I,J) = E12(I,J) * \text{HEFFM}(I,J) \]

The strain rates are returned to the FORM CSU.

### 3.3.5 CSC Ice Thickness Distribution

The Ice Thickness Distribution CSC accounts for the changes in ice thickness and concentration due to growth, advection, and deformation of the ice. Only thick and thin ice are considered for these calculations. The deformation of thick ice can create thin ice by divergence while the thin ice can be removed by convergence. Both growth and melt affect the amount of thick and thin ice. While growth can significantly decrease the amount of thin ice and increase the amount of thick ice, melting can add to the amount of thin ice or create open water by decreasing the thick ice. The advection of the ice thickness and compactness due to explicit time-stepping is done in the ADVECT CSU. The DIFFUS CSU is used to determine the diffusion of the ice thickness, compactness, and
concentration by using the explicit forward time differencing. The negative ice to be melted is calculated in the RNEG T CSU. The changes of thickness and compactness for each timestep are then estimated in the GROWTH CSU.

3.3.5.1 CSU ADVECT

The ADVECT CSU does the explicit time-stepping for the advection of the ice thickness and compactness within the ice model.

3.3.5.1.1 CSU ADVECT Design Specification/Constraints – There are no known constraints.

3.3.5.1.2 CSU ADVECT Design

Input Data Elements:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIFF1</td>
<td>harmonic diffusion constant</td>
</tr>
<tr>
<td>HEFF</td>
<td>mean ice thickness per grid cell or ice mass per grid area</td>
</tr>
<tr>
<td>HEFFM</td>
<td>thermodynamic land/sea mask</td>
</tr>
<tr>
<td>LAD</td>
<td>type of time finite difference (leapfrog or backward Euler)</td>
</tr>
<tr>
<td>UIICEC</td>
<td>intermediate x component of the ice drift</td>
</tr>
<tr>
<td>VICEC</td>
<td>intermediate y component of the ice drift</td>
</tr>
</tbody>
</table>

/STEP/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTAT</td>
<td>timestep (s)</td>
</tr>
<tr>
<td>DELTAX</td>
<td>x (longitude) grid spacing (deg)</td>
</tr>
<tr>
<td>DELTAY</td>
<td>y (latitude) grid spacing (deg)</td>
</tr>
</tbody>
</table>

/STEPSP/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTXA</td>
<td>x (longitude) grid spacing for thermodynamic fields (m)</td>
</tr>
<tr>
<td>DELTYA</td>
<td>y (latitude) grid spacing for thermodynamic fields (m)</td>
</tr>
</tbody>
</table>

Output Data Elements:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEFF</td>
<td>mean ice thickness per grid cell or ice mass per grid area</td>
</tr>
</tbody>
</table>

/DIFFU3/

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIFF3</td>
<td>harmonic diffusion constant</td>
</tr>
</tbody>
</table>

Parameters:

stored in file ice.par

The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Local Data Elements:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTT</td>
<td>timestep according to the type of time finite difference</td>
<td>real</td>
</tr>
<tr>
<td>DELTX</td>
<td>grid spacing in x direction</td>
<td>real</td>
</tr>
<tr>
<td>DELTY</td>
<td>grid spacing in y direction</td>
<td>real</td>
</tr>
<tr>
<td>DIFF2</td>
<td>harmonic diffusion constant</td>
<td>real</td>
</tr>
<tr>
<td>I</td>
<td>index counter</td>
<td>integer</td>
</tr>
<tr>
<td>J</td>
<td>index counter</td>
<td>integer</td>
</tr>
<tr>
<td>K3</td>
<td>index of time level</td>
<td>integer</td>
</tr>
</tbody>
</table>
**Software Test Description for PIPS2.0**

| KD       | loop index for diffusion (2) | integer |
| LL       | type of time finite difference | integer |
| RADIAN   | degrees per radian (57.29578°) | real    |
| RADIUS   | radius of the Earth (6370.0 × 10³ m) | real    |

**Algorithms:**

The advection terms are calculated by using the following:

\[
\left[ (uh)_x \right]_{i+1/2,j+1/2} = \frac{1}{4h} \left[ \left( h_{i+1/2,j+1/2} + h_{i+3/2,j+1/2} \right) \left( u_{i+1,j+1} + u_{i+1,j} \right) - \left( h_{i-1/2,j+1/2} + h_{i-3/2,j+1/2} \right) \left( u_{i-1,j+1} + u_{i-1,j} \right) \right],
\]

where \( h \) is the \( x \) (longitude) grid spacing and \( u \) is the \( x \) component of the ice-drift velocity. Solving for \( (vh)_y \), where \( h \) is the \( y \) (latitude) grid spacing and \( v \) is the \( y \) component of the ice-drift velocity, is done similarly, but

\[- \tan(lat) \times h \times v/r \quad (19)\]

is an extra term needed to transform from the Cartesian to spherical coordinates. \( lat \) is the latitude, \( h \) is the ice thickness or concentration, \( v \) is the ice drift along the \( y \) (latitude) axis, and \( r \) is the radius of the Earth.

**Data Conversions:**

The grid spacing, in degrees, of the latitudes must be converted to radians.

**Logic Flow:**

The ICEMDL CSU calls the ADVECT CSU.

Two types of time finite differences are allowed—backward Euler or leapfrog. For this version of the ice model, the backward Euler method is selected. For the backward Euler method, the timestep based on the type of time finite difference is set to the timestep (seconds) for the model run. If the leapfrog method were selected, the timestep based on the type of time finite difference would be two times the timestep (seconds) for the model run. The timestep index \( (K3) \) is set to 3 for the leapfrog method and 2 for the backward Euler method.

The mean ice thickness or ice mass per grid area (HEFF) is reordered so that the third time level is equal to the second time level and the second time level is equal to the first for each thermodynamic field:

\[
\text{HEFF}(I,J,3) = \text{HEFF}(I,J,2)
\]

\[
\text{HEFF}(I,J,2) = \text{HEFF}(I,J,1)
\]

Next, the process of calculating the standard conservative advection begins for each velocity field. First DELTX and DELTY are defined in the spherical coordinates with DELTX dependent on the latitude:
DELTX = DELTT/(4.0 * DELTAX),

DETY = DELTT/(4.0 * DELTAY),

where DELTT is the timestep according to the type of time finite difference, DELTAXA is the x (longitude) grid spacing in meters, DELTYA is the y (latitude) grid spacing in meters, and J is the index of the velocity field.

Then the advection term for all velocity fields at the first time level is calculated as:

\[ \text{HEFF}(I+1,J+1,1) = \text{HEFF}(I+1,J+1,K3) - \text{DELTX} \times \frac{1}{4} \times \left( \frac{\text{HEFF}(I+1,J+1,2)}{3} + \frac{\text{HEFF}(I+2,J+1,2)}{3} \right) \times (UICEC(I+1,J+1) + UICEC(I+1,J)) \\
- \frac{1}{2} \times \Delta Y \times \left( \frac{\text{HEFF}(I+1,J+1,1)}{2} + \frac{\text{HEFF}(I+1,J+1,2)}{2} \right) \times (UICEC(I,J+1) + UICEC(I,J)) \\
- \frac{1}{2} \times \Delta Y \times \left( \frac{\text{HEFF}(I+1,J+1,2)}{2} + \frac{\text{HEFF}(I+2,J+1,2)}{2} \right) \times (UICEC(I,J+1) + UICEC(I,J)) \\
- \frac{1}{2} \times \Delta Y \times \left( \frac{\text{HEFF}(I+1,J+1,1)}{2} + \frac{\text{HEFF}(I+1,J+1,2)}{2} \right) \times (UICEC(I,J) + UICEC(I+1,J)) \]

where UICEC is the intermediate x component of the ice drift and VICEC is the intermediate y component of the ice drift. Due to the transformation to spherical coordinates from the Cartesian coordinates, the following extra term is added:

\[ \text{HEFF}(I+1,J+1,1) = \text{HEFF}(I+1,J+1,1) \\
- \tan(\Delta Y) \times (J-MIDY-0.5)/RADIUS \times \text{HEFF}(I+1,J+1,2) \\
- \left( \frac{\text{ICEC}(I,J) + \text{ICEC}(I+1,J) + \text{ICEC}(I,J+1) + \text{ICEC}(I+1,J+1)}{4} \right) / RADIUS \]

where DELTAY is the y (latitude) grid spacing in degrees, RADIUS is the constant to convert from degrees to radians, VICEC is the ice-drift velocity along the y axis, and RADIUS is the radius of the Earth in meters.

If the backward Euler time finite difference is selected, the value of HEFF at each velocity field is corrected. First, the value of HEFF at the third time level is set equal to the value of HEFF at the second time level and the value of HEFF at the second time level is set to be the average of the values at the first and second time levels:

\[ \text{HEFF}(I,J,3) = \text{HEFF}(I,J,2) \]

\[ \text{HEFF}(I,J,2) = 0.5 \times (\text{HEFF}(I,J,1) + \text{HEFF}(I,J,2)) \]

The values of LL = 3 and K3 = 3 are initialized.

The standard conservative advection process is repeated. After completion, the value of HEFF at the second time level is set to be the value of HEFF at the third time level:

\[ \text{HEFF}(I,J,2) = \text{HEFF}(I,J,3) \]

Then HEFF is diffused at each velocity field for the third time level. Harmonic and biharmonic terms are computed. First, the value is set as:

\[ \text{DIFF}3(J) = \text{DIFF}1, \]

where DIFF3 is the biharmonic diffusion constant at each y of the thermodynamic field and DIFF1 is the harmonic diffusion constant.
HEFF is recalculated at the first time level for each velocity field before calling the DIFFUS CSU. HEFF is redefined as:

$$HEFF(I,J,1) = HEFF(I,J,1) + DELTT \times DIFF1 \times HEFFM(I,J)$$
$$\times (-\tan(DELTA*I/(J-MIDY-0.5)/\text{RADIUS}))$$
$$\times (HEFF(I,J+1,3) - HEFF(I,J-1,3))/2.0/DELTA*I/\text{RADIUS}. $$

Then the DIFFUS CSU is called to determine the diffusion of ice thickness or concentration. After returning from the DIFFUS CSU, the value of HEFF is recalculated for the first time level at the thermodynamic land/sea mask as

$$HEFF(I,J,1) = (HEFF(I,J,1) + HEFF(I,J,3)) \times HEFFM(I,J).$$

HEFF(I,J,3) from the previous call to the DIFFUS CSU now becomes the harmonic term. The values of the biharmonic term DIFF2 and the harmonic term (DIFF3) are defined in spherical coordinates:

DIFF2 = - DELTYA \times DELTYA/DELTET

DIFF3(J) = DIFF2

Before executing the DIFFUS CSU, HEFF is redefined as:

$$HEFF(I,J,1) = HEFF(I,J,1) - HEFFM(I,J)$$
$$\times (-\tan(DELTA*I/(J-MIDY-0.5)/\text{RADIUS}))$$
$$\times (HEFF(I,J+1,3) - HEFF(I,J-1,3))/2.0*(DELTA*I/\text{RADIUS}).$$

The DIFFUS CSU is called again to determine the diffusion of the ice thickness or concentration. After executing the DIFFUS CSU, the value of HEFF is recalculated for the first time level at the thermodynamic land/sea mask as:

$$HEFF(I,J,1) = (HEFF(I,J,1) + HEFF(I,J,3)) \times HEFFM(I,J).$$

Upon completion of this CSU, flow returns to the main program.

3.3.5.2 CSU DIFFUS

The purpose of the DIFFUS CSU is to determine the diffusion of the ice thickness and concentration by using explicit forward time differencing.

3.3.5.2.1 CSU DIFFUS Design Specification/Constraints – There are no known constraints.

3.3.5.2.2 CSU DIFFUS Design

Input Data Elements:
- DIFF1 harmonic diffusion constant
- HEFF mean ice thickness or concentration per grid cell on input; diffused value on output
- HEFFM thermodynamic land/sea mask

/DIFFUS3/
- DIFF3 harmonic diffusion constant
/STEP/
DELTA X  x (longitude) grid spacing (deg)
DELTA Y  y (latitude) grid spacing (deg)

/STEPSP/
DELTA X  x (longitude) grid spacing for thermodynamic fields (m)
DELTA Y  y (latitude) grid spacing for thermodynamic fields (m)

Output Data Elements:
HEFF   mean ice thickness or concentration per grid cell on input; diffused value on output

Parameters:
stored in file ice.par

The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Local Data Elements:
DELTA X  x grid spacing based on timestep, grid spacing, and diffusion constant
DELTA Y  y grid spacing based on timestep, grid spacing, and diffusion constant
HEFF I  diffused ice thickness or concentration
I     index counter
J     index counter

Logic Flow:
The DIFFUS CSU is called by the ADVECT CSU to determine the diffusion of ice thickness or concentration. All values of HEFF I are initialized to 0. Then, DELTA X and DELTA Y are calculated for each row of velocity fields.

\[
\text{DELTA X} = \text{DELT T} \times \text{DIFF I} / (\text{DELTA X} \times 2),
\]
\[
\text{DELTA Y} = \text{DELT T} \times \text{DIFF I} / (\text{DELTA Y} \times 2),
\]

where DELT T is the timestep, DIFF I is the harmonic diffusion constant, DIFF 3 is the diffusion constant, DELTA X is the x (longitude) grid spacing in degrees and DELTA Y is the y (latitude) grid spacing in degrees. Now the diffused value of the ice thickness or concentration is calculated by:

\[
\begin{align*}
\text{HEFF I} (I,J) &= \text{DELTA X} \times (\text{HEFF I} (I+1,J,3) & - \text{HEFF I} (I,J,3)) \times \text{HEFF M} (I+1,J) \\
&\quad - (\text{HEFF I} (I,J,3) & - \text{HEFF I} (I-1,J,3)) \times \text{HEFF M} (I-1,J) \\
&\quad + \text{DELTA Y} \times ((\text{HEFF I} (I,J+1,3) & - \text{HEFF I} (I,J,3)) \times \text{HEFF M} (I,J+1) \\
&\quad - (\text{HEFF I} (I,J,3) & - \text{HEFF I} (I,J-1,3)) \times \text{HEFF M} (I,J-1)).
\end{align*}
\]

The values of each thermodynamic field at the third time level is then set equal to the diffused value.

\[
\text{HEFF I} (I,J,3) = \text{HEFF I} (I,J)
\]
3.3.5.3 CSU GROWTH

The GROWTH CSU calculates the change of thickness and compactness for each timestep.

3.3.5.3.1 CSU GROWTH Design Specification/Constraints – There are no known constraints.

3.3.5.3.2 CSU GROWTH Design

Input Data Elements:
- A22: minimum ice concentration
- FO: growth rate of thin ice
- HDIFF1: net growth of thin ice
- HEFFM: land/sea mask for thermodynamic fields
- HO: minimum ice thickness
- OUT: land/sea mask including outflow conditions for thermodynamic variables

/GROW/
- FHEFF: total growth rate of thick ice

/RSTRT/
- AREA1: fraction of grid cell covered by ice
- HEFF: mean ice thickness per grid cell

/STEP/
- DELTAT: timestep (s)

Output Data Elements:
- FO: growth rate of thin ice
- GAREA: change of areal ice extent due to melting and freezing
- HCORR: additional ice to be melted for the mixed-layer balance
- HDIFF1: net growth of thin ice

/GROW/
- FHEFF: total growth rate of thick ice

/RSTRT/
- AREA1: fraction of grid cell covered by ice
- HEFF: mean ice thickness per grid cell

Parameters:
stored in file ice.par

The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Local Data Elements:
- GHEFF(IMT,JMT): average growth tendency for the timestep, real
- I: index counter, integer
- J: index counter, integer
Logic Flow:
The GROWTH CSU is called by the ICEMDL CSU.

Upon entering this CSU, the growth rates of thick and thin ice are used to calculate the total change of ice thickness and concentration in a grid cell. To do this, the average growth tendency of this timestep (GHEFF), the average compactness tendency for this timestep (GAREA), the net growth of thin ice for this timestep (HDIFF1), and the additional ice to be melted for the mixed-layer balance (HCORR) are calculated first:

\[
\begin{align*}
GHEFF(I,J) &= -DELTAT \times FHEFF(I,J), \\
GAREA(I,J) &= DELTAT \times FO(I,J), \\
GHEFF(I,J) &= -1.0 \times \min(HEFF(I,J,1), GHEFF(I,J)), \\
HDIFF1(I,J) &= -DELTAT \times HDIFF1(I,J), \\
GAREA(I,J) &= \max(0.0, GAREA(I,J)), \\
HDIFF1(I,J) &= -1.0 \times \min(HEFF(I,J,1), HDIFF1(I,J)), \\
HCORR(I,J) &= \min(0.0, GHEFF(I,J)),
\end{align*}
\]

where DELTAT is the timestep in seconds, FHEFF is the total growth rate of thick ice, FO is the growth rate of thin ice, and HEFF is the mean ice thickness per grid cell.

The value of GAREA must not be less than 0. Then the change of areal ice extent due to melting and freezing is calculated by:

\[
GAREA(I,J) = 2.0 \times (1.0 - AREA1(I,J,2)) \times GAREA(I,J)/HO + 0.5 \times HCORR(I,J) \times AREA1(I,J,2)/HEFF(I,J,1) + 0.00001,
\]

where AREA1 is the fraction of the grid cell covered by ice and HO is the demarcation between thick and thin ice. The values of AREA1 and HEFF are corrected by:

\[
\begin{align*}
AREA1(I,J,1) &= AREA1(I,J,1) + GAREA(I,J), \\
HEFF(I,J,1) &= HEFF(I,J,1) + GHEFF(I,J) \times OUT(I,J),
\end{align*}
\]

where OUT is the land/sea mask including outflow conditions for thermodynamic variables. The value of HCORR is recalculated by:

\[
HCORR(I,J) = GHEFF(I,J) - DELTAT \times FHEFF(I,J).
\]

The RNEGT CSU is called to remove the "negative" ice from the ice thickness field.

The outside points in the AREA1 and HEFF arrays are zeroed out by multiplying the current value by the land/sea mask value for each point

\[
\begin{align*}
AREA1(I,J,1) &= AREA1(I,J,1) \times HEFFM(I,J) \\
HEFF(I,J,1) &= HEFF(I,J,1) \times HEFFM(I,J).
\end{align*}
\]

If there is no ice at a point, then the value of AREA1 is set to 0. This new value of AREA1 is checked again to make sure that it is not greater than 1 and is not less than the minimum
compactness allowed. The amount of additional ice to be melted for the mixed-layer balance (HCORR) is calculated by:

\[ \text{HCORR}(I,J) = \text{GHEFF}(I,J) - \text{DELTAT} \times \text{FHEFF}(I,J). \]

The values of total growth rate of the thick ice (FHEFF) and the growth rate of thin ice (FO) are stored.

The ice thickness, ice concentration, net growth rate of thin ice, growth rate of thin ice, the negative ice to be melted, and the total growth rate of thick and thin ice are returned to the main driving CSU.

3.3.5.4 CSU RNEGRT

The RNEGRT CSU removes the negative ice from the model. This method essentially removes the amount of thin ice due to converging conditions.

3.3.5.4.1 CSU RNEGRT Design Specification/Constraints – There are no known constraints.

3.3.5.4.2 CSU RNEGRT Design

Input Data Elements:
- FH  additional ice to be melted
- HEFF mean ice thickness per grid cell

Output Data Elements:
- HEFF mean ice thickness per grid cell

Parameters:
stored in file ice.par

The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Local Data Elements:
- I index counter  integer
- J index counter  integer

Logic Flow:
The RNEGRT CSU is called by the GROWTH CSU.

The melt value is subtracted from the ice thickness value at the first time level (HEFF) and assigned to the ice thickness value at the third time level:

\[ \text{HEFF}(I,J,3) = \text{HEFF}(I,J,1) - \text{FH}(I,J). \]

The ice thickness value is tested to make certain that it is not less than 0. The new value of HEFF is returned to the GROWTH CSU.
3.3.6 CSC Ice Strength

The ice strength, as a function of ice thickness distribution, is calculated in this CSU. The strength of the ice depends on the amount of thin ice and is calculated in the FORM CSU.

3.3.6.1 CSU XSUM

The purpose of the XSUM CSU is to sum every value of the input array into a scalar.

3.3.6.1.1 CSU XSUM Design Specification/Constraints – There are no known constraints.

3.3.6.1.2 CSU XSUM Design

Input Data Elements:
- HEFF mean ice thickness per grid cell

Output Data Elements:
- S1 summation of mean ice thickness values

Parameters:
- stored in file ice.par
  The ice.par parameters are described in Sec. 3.1.1.2.

  Data element design information is provided in Sec. 4.0.

Local Data Elements:
- I index counter integer
- J index counter integer

Algorithms:
When calculating the ice thickness SUM, use

$$SUM = \sum_{j=1}^{ny} \sum_{i=1}^{nx} HEFF_{ij},$$  \hspace{1cm} (20)

where $HEFF$ is the ice thickness per grid cell, $ny$ is the maximum number of columns of thermodynamic fields allowed, and $nx$ is the maximum number of rows of thermodynamic fields allowed.

Logic Flow:
CSU XSUM is called by the ICEMDL CSU to find the total ice in the basin.

The total basin ice thickness is calculated by summing the mean ice thickness for all thermodynamic fields.

The summation of the mean ice thickness values is returned to the main driving program.

3.3.6.2 CSU FORM

See Sec. 3.3.3.1 for a detailed description of the FORM CSU.
3.3.7 CSC Heat Balance

The Heat Balance CSC is used to calculate the heat budget. The HEAT CSU calculates the terms needed and the BUDGET CSU uses these terms to compute the growth rates of thick and thin ice.

3.3.7.1 CSU HEAT

The HEAT CSU takes the FNMOC forcing and calculates terms needed for the heat budget balance.

3.3.7.1.1 CSU HEAT Design Specification/Constraints – There are no known constraints.

3.3.7.1.2 CSU HEAT Design

**Input Data Elements:**

- **FO**
  - growth rate of ice on open water

- **/COX2/**
  - **CFO**
    - either the heat above the freezing temperature in terms of ice thickness growth rate or the ice thickness growth rate in open water
  - **TFRZ**
    - temperature at the freezing point (Kelvin)
  - **TM1**
    - mixed-layer temperature from the previous timestep (Kelvin)

- **/OCEANS/**
  - **FW**
    - oceanic heat flux

- **/RAD/**
  - **FSH**
    - atmospheric forcing – solar radiation

- **/RFOR/**
  - **ES**
    - NOGAPS atmospheric forcing – surface vapor pressure
  - **ES1**
    - NOGAPS atmospheric forcing – sensible heat flux
  - **PS**
    - NOGAPS atmospheric forcing – surface air pressure
  - **PS1**
    - NOGAPS atmospheric forcing – total heat flux
  - **TAIR**
    - NOGAPS atmospheric forcing – surface air temperature

- **/RFOR2/**
  - **GAIRX**
    - x component of the geostrophic wind
  - **GAIRY**
    - y component of the geostrophic wind

- **/RSTRT/**
  - **AREA1**
    - fraction of the grid cell covered by ice
  - **HEFF**
    - mean ice thickness per grid cell
  - **TICE**
    - mixed-layer temperature for open water or ice temperature for ice-covered water

**Output Data Elements:**

- **FO**
  - growth rate of ice on open water
- **HDIFF1**
  - net growth of thin ice
- **SHICE**
  - total ice thickness

- **/COX2/**
  - **CFO**
    - either the heat above the freezing in terms of ice thickness growth rate or the ice thickness growth rate in open water
FW1: heat above the freezing temperature
GICE: ice thickness growth rate of open water

/GROW/
FHEFF: total growth rate of thick ice

/RFOR/
TAIR: NOGAPS atmospheric forcing – surface air temperature

/RSTRT/
AREA1: fraction of the grid cell covered by ice
TICE: mixed-layer temperature for open water or ice temperature for ice-covered water

Parameters:
stored in file ice.par
The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Local Data Elements:
A22: minimum ice concentration 
real
AR(IMIT,JMT): used in determining the amount of open area 
real
CFO_TMP: temporary storage for heat flux 
real
CP: conversion factor from joules to calories (4.19 × 10^6) 
real
EXCHNG: frequency of exchanging ice/ocean data 
real
FLO(IMITP1,JMTP1): net longwave radiation 
real
GX: x component of the geostrophic wind speed 
real
GY: y component of the geostrophic wind speed 
real
HICE(IMIT,JMT): average ice thickness over the grid cell 
real
I: index counter 
integer
IMT: total number of T grid boxes zonally (360) 
integer
J: index counter 
integer
JMT: total number of T grid boxes meridionally (360) 
integer
KOPEN: flag for open water or ice-covered water calculations 
integer
QA(IMITP1,JMTP1): specific humidity at the ice surface 
real
RLATNT: volumetric heat of fusion of ice (302 × 10^6 joules) 
real
TDAY: timestep to exchange ice-ocean conditions 
real
TMIX(IMIT,JMT): mixed-layer temperature 
real
UG(IMIT,JMT): magnitude of wind 
real
ZMIX: mixed-layer thickness (30 m) 
real

Algorithms:
The surface heat budget (Parkinson and Washington 1979; Manabe et al. 1979) is calculated using the following equation:

\[ (1 - \alpha) F_s + F_L + D_1 \left[ \mathbf{U}_g \right] (T_a - T_0) + D_2 \left[ \mathbf{U}_g \right] \left[ q_a(T_a) - q_s(T_0) \right] - D_3 T_0^4 + (K/H)(T_w - T_0) = 0, \quad (21) \]

where \( \alpha \) is the surface albedo, \( T_0 \) is the surface temperature of ice, \( T_a \) is the air temperature, \( T_w \) is the water temperature, \( \mathbf{U}_g \) is the geostrophic wind, \( q_a \) is the specific humidity of air, \( q_s \) is the
specific humidity of the ice surface, $F_s$ is the incoming short-wave radiation, $F_L$ is the incoming longwave radiation, $D_1$ is the bulk sensible heat transfer coefficient, $D_2$ is the bulk latent heat transfer coefficient (water or ice), $D_3$ is the Stefan-Boltzmann constant times the surface emissivity, $K$ is the ice conductivity, and $H$ is the ice thickness.

**Logic Flow:**

The HEAT CSU is called by the ICEMDL CSU.

To begin the process of solving the equation for the surface heat budget, the $x$ (GX) and $y$ (GY) components of geostrophic wind speed are calculated for each thermodynamic field. These $x$ and $y$ components, which are an average of the wind speed at the corners of the grid cell, are then used to find the magnitude of the geostrophic wind (UG):

$$
GX = (GAIRX(I,J) + GAIRX(I+1,J) + GAIRX(I,J+1) + GAIRX(I+1,J+1)) \times 0.25
$$

$$
GY = (GAIRY(I,J) + GAIRY(I+1,J) + GAIRY(I,J+1) + GAIRY(I+1,J+1)) \times 0.25
$$

$$
UG(I,J) = \sqrt{(GX \times 2 + GY \times 2)}.
$$

Next, the specific humidity at the ice surface (QA) for each thermodynamic field is calculated from the surface vapor pressure and the surface pressure fields of atmospheric forcing.

$$
QA(I,J) = (0.622 \times ES(I,J))/(PS(I,J) - ES(I,J)),
$$

where $ES$ is the surface vapor pressure and $PS$ is the surface air pressure.

The net long wave radiation (FLO) is calculated using the total heat flux minus the sensible heat flux plus the solar radiation. The sign of the net long wave radiation is then changed to correct the net long wave error.

$$
FLO(I,J) = PS1(I,J) - ES1(I,J) + FSH(I,J)
$$

$$
FLO(I,J) = - FLO(I,J),
$$

where $PS1$ is the total heat flux, $ES1$ is the sensible heat flux and $FSH$ is the solar radiation.

The area of the grid cell covered by thick ice cannot be <0.15; each value at the second time level is checked to make sure this does not happen.

Next, the total growth rate of thick ice (FHEFF) is set to 0, the average ice thickness over the grid cell (HICE) is calculated, and the term to be used in determining the amount of ice (AR) is determined for each thermodynamic field:

$$
FHEFF(I,J) = 0,
$$

$$
HICE(I,J) = HEFF(I,J,2)/AREA1(I,J,2),
$$

$$
AR(I,J) = \text{MIN}(\text{AREA1}(I,J,2), \text{HEFF}(I,J,2) \times 10000).
$$

Constant values are assigned for the volumetric heat of fusion (RLATNT), the conversion factor from joules to calories (CP), and the thickness of the mixed layer (ZMIX). The exchange of
ice and ocean conditions (EXCHNG) is set to every 3 h or 8 times per day, and the number of seconds within this exchange time (TDAY) is also set:

\[
\begin{align*}
RLATNT &= 302.0 \times 10^6 \\
CP &= 4.19 \times 10^6 \\
ZMIX &= 30.0 \\
EXCHNG &= 8.0 \\
TDAY &= 86400.0/EXCHNG.
\end{align*}
\]

The value of T MIX is reset to the mixed-layer temperature (Kelvin) from the previous timestep for each thermodynamic field:

\[
TMIX(I,J) = TM1(I,J).
\]

The value of KOPEN is set to –1 for the open water case and the BUDGET CSU is called to calculate the change in growth rate for open water.

Next, the value of KOPEN is set to 2 for an ice-covered ocean case and the BUDGET CSU is called again, but this time it calculates the change in growth rate for ice-covered water.

After executing from the BUDGET CSU for the second time, the fractional rate of total open water growth (HDIFF1) per grid cell, the net rate of total open water growth rate (FW1), the heat flux (CFO_TMP), and the heat above the freezing in terms of ice thickness growth rate (CFO) are calculated:

\[
\begin{align*}
HDIFF1(I,J) &= (1.0 - AR(I,J)) \times FO(I,J) \\
FW1(I,J) &= FO(I,J) \\
CFO_TMP &= CP \times ZMIX \times (TM1(I,J)-TFRZ(I,J))/TDAY/RLATNT \\
CFO(I,J) &= CFO_TMP \times (1.0-AR(I,J)).
\end{align*}
\]

After these values are calculated, quality checks are performed. If the calculated value of CFO is <0, then it is set to 0. If the calculated value of the ice thickness growth rate HDIFF1 <0, then HDIFF1 is not changed since the ice thickness has not increased. The ice is considered to have melted for this case. If HDIFF1 ≥0 and HDIFF1 ≥CFO, then the ice grows in the open water. If HDIFF1 ≥0 and HDIFF1 <CFO, then the ice is stable; i.e., the ice is neither grown nor melted. For this case, the water is cooled but the ice does not grow because the atmospheric cooling cannot take away all of the heat above the freezing temperature. HDIFF1 and FW1 are reset to 0, i.e., there are no ice growth rates.

If (HDIFF1(I,J) ≥0.0), then

if (HDIFF1(I,J) ≥CFO(I,J)), then

\[
\begin{align*}
HDIFF1(I,J) &= HDIFF1(I,J) - CFO(I,J) \\
FW1(I,J) &= FO(I,J) - CFO_TMP,
\end{align*}
\]

else

\[
\begin{align*}
HDIFF1(I,J) &= 0 \\
FW1(I,J) &= 0.
\end{align*}
\]
The ice thickness growth rate of open water and the total ice thickness are calculated by:

\[
\text{GICE}(I,J) = (1.0 - \text{AR}(I,J)) \times (\text{FO}(I,J) + \text{FW}(I,J) \times 10^{-6}/302.0), \\
\text{SHICE}(I,J) = \text{FHEFF}(I,J) \times \text{AR}(I,J) + \text{HDIFF}(I,J).
\]

The values of FHEFF, FO, and FW1 are redefined as:

\[
\text{CFO}(I,J) = (1.0 - \text{AR}(I,J)) \times \text{FO}(I,J), \\
\text{FHEFF}(I,J) = \text{FHEFF}(I,J) \times \text{AR}(I,J) + \text{HDIFF}(I,J), \\
\text{FO}(I,J) = \text{FW1}(I,J), \\
\text{FW1}(I,J) = \text{CP} \times \text{ZMIX} \times (\text{TM1}(I,J) - \text{TFRZ}(I,J)) \times (1.0 - \text{AR}(I,J))/\text{TDAV}/\text{RLATNT}, \\
\text{FW1}(I,J) = \text{MAX}(0, \text{FW1}(I,J)).
\]

The program flow returns to the ICEMDL CSU.

**3.3.7.2 CSU BUDGET**

The purpose of the BUDGET CSU is to compute the growth rates of thick and thin ice. It also computes the surface temperature of ice by iteration. This temperature balances the surface heat budget and dictates the conduction of heat through the ice and, therefore, the growth rates.

3.3.7.2.1 CSU BUDGET Design Specification/Constraints – There are no known constraints.

3.3.7.2.2 CSU BUDGET Design

**Input Data Elements:**

- **FLO** longwave radiation
- **HICE1** average ice thickness over the grid cell
- **KOPEN** flag for ice-covered or open water
- **QA** specific humidity at the ice surface
- **TAIR** NOGAPS atmospheric forcing – surface air temperature
- **TSFC** mixed-layer temperature for open water or ice temperature for ice-covered water (Kelvin)
- **UG** magnitude of the wind

/OCEANS/

- **FW** oceanic heat flux

/RAD/

- **FSH** atmospheric forcing – solar radiation

/SNOW/

- **SNRT** snowfall rate

/STEP/

- **DELTAT** timestep (s)

**Output Data Elements:**

- **TSFC** mixed-layer temperature for open water or ice temperature for ice-covered water (Kelvin)
- **TSUM** mean ice thickness from all seven levels
Parameters:
stored in file ice.par

The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Local Data Elements:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1(IMT,JMT)</td>
<td>fixed forcing term in heat budget</td>
<td>real</td>
</tr>
<tr>
<td>A2(IMT,JMT)</td>
<td>used in heat budget</td>
<td>real</td>
</tr>
<tr>
<td>A3(IMT,JMT)</td>
<td>used in heat budget</td>
<td>real</td>
</tr>
<tr>
<td>AKI</td>
<td>ice conductivity constant for the ice cover case (2.1656 W m⁻¹K⁻¹)</td>
<td>real</td>
</tr>
<tr>
<td>AKS</td>
<td>snow conductivity constant for the ice cover case (0.31)</td>
<td>real</td>
</tr>
<tr>
<td>B(IMT,JMT)</td>
<td>used in heat budget</td>
<td>real</td>
</tr>
<tr>
<td>D1</td>
<td>bulk sensible heat transfer coefficient (2.284 J/m³ K)</td>
<td>real</td>
</tr>
<tr>
<td>D1I</td>
<td>bulk latent heat transfer coefficient over ice (6.4475 × 10³ J/m³)</td>
<td>real</td>
</tr>
<tr>
<td>D1W</td>
<td>bulk latent heat transfer coefficient over water (5.6875 × 10³ J/m³)</td>
<td>real</td>
</tr>
<tr>
<td>D3</td>
<td>Stefan-Boltzmann constant times the surface emissivity (5.5 × 10⁻⁸ W/m² K⁴)</td>
<td>real</td>
</tr>
<tr>
<td>EM(IMT,JMT)</td>
<td>used in computing terms in the heat budget for an ice-covered case</td>
<td>real</td>
</tr>
<tr>
<td>FICE(IMT,JMT)</td>
<td>ice growth at each of the seven levels</td>
<td>real</td>
</tr>
<tr>
<td>FW2(IMT,JMT)</td>
<td>oceanic heat flux (W/m²)</td>
<td>real</td>
</tr>
<tr>
<td>HICE(IMT,JMT)</td>
<td>ice depth at each of the seven levels</td>
<td>real</td>
</tr>
<tr>
<td>HS1(IMT,JMT)</td>
<td>snow depth at each of the seven levels</td>
<td>real</td>
</tr>
<tr>
<td>I</td>
<td>index counter</td>
<td>integer</td>
</tr>
<tr>
<td>IMAX</td>
<td>maximum number of iterations (10)</td>
<td>integer</td>
</tr>
<tr>
<td>ITER</td>
<td>counter for iterations</td>
<td>integer</td>
</tr>
<tr>
<td>J</td>
<td>index counter</td>
<td>integer</td>
</tr>
<tr>
<td>NLEVEL</td>
<td>number of levels (7)</td>
<td>integer</td>
</tr>
<tr>
<td>NLV</td>
<td>index of loop over number of levels</td>
<td>integer</td>
</tr>
<tr>
<td>NZ3</td>
<td>used to find the middle level</td>
<td>integer</td>
</tr>
<tr>
<td>POOL1(IMT)</td>
<td>temporary array</td>
<td>real</td>
</tr>
<tr>
<td>POOL2(IMT)</td>
<td>temporary array</td>
<td>real</td>
</tr>
<tr>
<td>Q0(IMT,JMT)</td>
<td>the inverse of volumetric heat of fusion of ice (m³/j)</td>
<td>real</td>
</tr>
<tr>
<td>QIS</td>
<td>used in equation for determining excess heat after all snow is melted (0.364)</td>
<td>real</td>
</tr>
<tr>
<td>QS1</td>
<td>used to calculate terms in heat budget (6.1402 × 10⁻⁴)</td>
<td>real</td>
</tr>
<tr>
<td>SNOW(IMT,JMT)</td>
<td>stores midpoint value of snow depth at all levels</td>
<td>real</td>
</tr>
<tr>
<td>SSUM(IMT,JMT)</td>
<td>mean of snow depth from all seven levels</td>
<td>real</td>
</tr>
<tr>
<td>TB</td>
<td>freezing point of seawater (271.2 K)</td>
<td>real</td>
</tr>
<tr>
<td>THICK(IMT,JMT)</td>
<td>midpoint value of ice thickness</td>
<td>real</td>
</tr>
<tr>
<td>TICE(IMT,JMT)</td>
<td>temperature of mixed layer for open water case; ice temperature for ice-covered case</td>
<td>real</td>
</tr>
<tr>
<td>TMELT</td>
<td>freezing point (273.16 K)</td>
<td>real</td>
</tr>
<tr>
<td>TMID(IMT,JMT)</td>
<td>midpoint surface temperature</td>
<td>real</td>
</tr>
<tr>
<td>TMID2(IMT,JMT)</td>
<td>midpoint surface temperature of the next timestep</td>
<td>real</td>
</tr>
<tr>
<td>TX1</td>
<td>ice constant</td>
<td>real</td>
</tr>
<tr>
<td>TX2</td>
<td>snow constant</td>
<td>real</td>
</tr>
<tr>
<td>TY</td>
<td>used in computing snow and ice thickness</td>
<td>real</td>
</tr>
<tr>
<td>TY1</td>
<td>ice constant</td>
<td>real</td>
</tr>
<tr>
<td>TZ</td>
<td>surface ice temperature at each level (°C)</td>
<td>real</td>
</tr>
</tbody>
</table>
TZ1  ice constant  real
TZ2  snow constant  real
V1  used to correct the snow depth  real

**Algorithms:**

The thermal conductivity is a single value based on a weighted sum of snow and ice conductivities:

\[
\frac{K_s K_I}{K_s S n_{LVL} + K_I h_{LVL}},
\]

where \( K_s \) is the snow conductivity, \( K_I \) is the ice conductivity, \( S n_{LVL} \) is the snow depth at the current level, and \( h_{LVL} \) is the ice thickness at the same level.

The surface heat budget (Parkinson and Washington 1979; Manabe et al. 1979) is calculated by using:

\[
(1 - \alpha) F_s + F_L + D_1 \left[ \bar{U}_g \right] (T_a - T_0) + D_2 \left[ \bar{U}_g \right] [q_a(T_a) - q_s(T_0)] - D_3 T_0^4
+ (K/H)(T_w - T_0) = 0,
\]

where \( \alpha \) is the surface albedo, \( T_0 \) is the surface temperature of ice, \( T_w \) is the air temperature, \( T_a \) is the water temperature, \( \bar{U}_g \) is the geostrophic wind, \( q_a \) is the specific humidity of air, \( q_s \) is the specific humidity of the ice surface, \( F_s \) is the incoming short-wave radiation, \( F_L \) is the incoming longwave radiation, \( D_1 \) is the bulk sensible heat transfer coefficient, \( D_2 \) is the bulk latent heat transfer coefficient (water or ice), \( D_3 \) is the Stefan-Boltzmann constant times the surface emissivity, \( K \) is the ice conductivity, and \( H \) is the ice thickness.

**Logic Flow:**

The HEAT CSU calls the BUDGET CSU.

In BUDGET CSU, the constant QS1, the freezing point of seawater TB, the maximum number of iterations IMAX, the bulk heat transfer coefficient D1, the bulk latent heat transfer coefficient over water D1W, the bulk latent heat transfer coefficient over ice D1I, the Stefan-Boltzmann constant D3, and the freezing point TMELT are defined:

\[
\begin{align*}
QS1 &= 0.622/1013.0 \\
IMAX &= 10.0 \\
TB &= 271.2 \\
D1 &= 2.284 \\
D1W &= 5.6875 \times 10^3 \\
D1I &= 6.4475 \times 10^3 \\
D3 &= 5.5 \times 10^{-8} \\
TMELT &= 273.16.
\end{align*}
\]

The initial snow depth HSI is set to 0 at each of the thermodynamic fields.
The oceanic heat flux is computed for each thermodynamic field as

\[ \text{FW2}(I,J) = \text{FW}(I,J). \]

For the ice-covered case, some initial conditions are assigned at each thermodynamic field:

\begin{align*}
\text{EM}(I,J) & = 0.99 \\
\text{HICE}(I,J) & = 0.80 \\
\text{POOL1}(I) & = 0.97 \\
Q0(I,J) & = (1.0 \times 10^{-6})/110.0,
\end{align*}

where EM, HICE, and POOL1 are all values used in computing the heat budget, and Q0 is the inverse of the volumetric heat of fusion of ice. Next, the value of the snow depth (HSI) is checked and the value of EM is reassigned accordingly:

\[ \text{if (HSI}(I,J) = 0), \text{then set EM}(I,J) = \text{POOL1}(I). \]

Each value of POOL1 is reassigned, the value of HICE is checked and the value of Q0 is reassigned if the snow depth (HSI) is equal to 0.

\begin{align*}
\text{POOL1}(I) & = (1.0 \times 10^{-6})/302.0 \\
\text{HICE}(I,J) & = \text{MAX}(\text{HICE1}(I,J), 0.05) \\
\text{if (HSI}(I,J) = 0), \text{then set Q0}(I,J) = \text{POOL1}(I).
\end{align*}

Next, the fixed forcing term in the heat budget can be determined for the case of ice covered

\[ A1(I,J) = \text{FSH}(I,J) + \text{FLO}(I,J) + D1 * \text{UG}(I,J) * \text{TAIR}(I,J) + D11 * \text{UG}(I,J) * \text{QA}(I,J), \]

where FSH is the solar radiation, UG is the geostrophic wind, TAIR is the surface air temperature and QA is the specific humidity at the ice surface.

For the ice-covered case, the growth at the seven levels is calculated after some initial conditions are set up at each thermodynamic field for the midpoint value of ice thickness THICK, the midpoint value of snow depth SNOW, the mean ice thickness TSUM, the mean snow depth SSUM, and the midpoint surface temperature TMID:

\begin{align*}
\text{THICK}(I,J) & = \text{HICE}(I,J) \\
\text{SNOW}(I,J) & = \text{HSI}(I,J) \\
\text{TSUM}(I,J) & = 0 \\
\text{SSUM}(I,J) & = 0 \\
\text{TMID}(I,J) & = \text{TSFC}(I,J).
\end{align*}

Now the snow and ice thickness (HSI and HICE, respectively) are calculated for the current level at each thermodynamic field:

\begin{align*}
\text{TSFC}(I,J) & = \text{TMID}(I,J) \\
\text{HICE}(I,J) & = \text{NLV} * 2.0 * \text{THICK}(I,J)/(\text{NLEVEL} + 1)
\end{align*}
TZ = TSFC(I,J) − 273.16
TY = NLV * 2.0 * SNOW(I,J)/(NLEVEL+1),

where NLV is the index of the current level and NLEVEL is the total number of levels to be processed. The value of HSI is reassigned based on the value of TZ:

if (TZ < 0) then HSI(I,J) = TY
if (TZ ≥ 0) then HSI(I,J) = SNOW(I,J).

The terms in the heat budget are computed using an iterative process. The iteration is started by calculating the following for each x and y thermodynamic field:

\[ B(I,J) = QS1 * 6.11 * \exp(21.8746 * (TSFC(I,J) – TMELT)/(TSFC(I,J) – TMELT + 265.5)) \]

\[ A3(I,J) = D1I * UG(I,J) * B(I,J) * 21.8746 * 265.5/((TSFC(I,J)–TMELT+265.5) * 2) \]

\[ A2(I,J) = − D1 * UG(I,J) * TSFC(I,J) – D1I * UG(I,J) * B(I,J) \]

\[ TICE(I,J) = AKI * AKS/(AKS * HICE(I,J) + AKI * HSI(I,J)) \]

\[ A3(I,J) = A3(I,J) + 4.0 * D3 * EM(I,J) * TSFC(I,J) * 3 + TICE(I,J) + D1 * UG(I,J) \]

\[ B(I,J) = TICE(I,J) * (TB – TSFC(I,J)). \]

If the maximum number of iterations has not been reached, then the surface temperature of ice TSFC is determined at each x and y thermodynamic field:

\[ TSFC(I,J) = TSFC(I,J) + (A1(I,J) + A2(I,J) + B(I,J))/A3(I,J), \]

If (TSFC(I,J) ≤ 200.0 and TAIR(I,J) ≥ TMELT),
then TSFC(I,J) = TMELT

If (TSFC(I,J) ≤ 200.0 and TAIR(I,J) < TMELT)
then TSFC(I,J) = 200.0.

The iteration counter is incremented and then checked to see if the maximum number of iterations has been reached. If it has, any value of the surface temperature of ice that is above freezing is reset to the freezing point (273.16 K). The iterative process continues until the maximum number of iterations has been computed.

After the maximum number of iterations has been computed, the ice temperature (TICE) at each x and y thermodynamic field is calculated:

\[ TICE(I,J) = TICE(I,J) * (HICE(I,J)/AKI * TSFC(I,J) + HSI(I,J)/AKS * TB) \]

If |HSII| < 0.0001, it is set to 0.0001.
The depth of melted snow (POOL1) and the excess heat after all snow is melted (POOL2) are then calculated.

\[
\text{POOL1}(t) = Q_0(I,J) \ast (A_1(I,J) + A_2(I,J) + \text{AKS/HSI}(I,J) \ast (\text{TICE}(I,J) - \text{TSFC}(I,J)))
\]

\[
\text{POOL2}(t) = (\text{HSI}(I,J)/\text{DELTAT} - \text{POOL1}(I)) \ast QIS \\
+ 1.0 \ast 10^{-6}/302.0 \ast (\text{AKS/HSI}(I,J) \ast (\text{TICE}(I,J) - \text{TSFC}(I,J))) \\
- \text{AKI/HICE}(I,J) \ast (\text{TB} - \text{TICE}(I,J)) \\
+ 0.09 \ast \text{FSH}(I,J) + 0.02 \ast \text{FLO}(I,J))
\]

If snow existed and POOL2 < 0, then the ice is melted. If no snow existed, POOL1 is recalculated using ice constants, instead of snow constants, to obtain the ice melt:

\[
\text{TZ1} = \text{HSI}(I,J) - 0.0001
\]

\[
\text{TY1} = - \text{POOL1}(I) - (\text{AKI/HICE}(I,J) \ast (\text{TB} - \text{TICE}(I,J)) \\
- \text{AKS/HSI}(I,J) \ast (\text{TICE}(I,J) - \text{TSFC}(I,J))) \ast Q_0(I,J)
\]

\[
\text{TX1} = \text{MIN}(0.0, \text{POOL2}(I))
\]

if (TZ1 = 0), then \(\text{FICE}(I,J) = \text{TY1}\)

if (TZ1 = 0), then \(\text{FICE}(I,J) = \text{TX1}\)

if (TZ1 = 0), then \(\text{HSI}(I,J) = 0\).

The snow depth is corrected by:

\[
\text{V1} = \text{HSI}(I,J) - \text{POOL}(I) \ast \text{DELTAT}
\]

\[
\text{POOL1}(I) = \text{MAX}(V1, 0.0)
\]

\[
\text{TZ2} = \text{TSFC}(I,J) - \text{TMELT}
\]

\[
\text{TX2} = \text{HSI}(I,J) + \text{DELTAT} \ast \text{SNRT}
\]

if (TZ2 = 0), then \(\text{HSI}(I,J) = \text{POOL1}(I)\)

if (TZ2 = 0), then \(\text{HSI}(I,J) = \text{TX2}\).

The ice melt at the ice-ocean interface is added to the melt at the top of the ice by:

\[
\text{FICE}(I,J) = 1.0 \ast 10^{-6}/302.0 \ast (\text{AKI/HICE}(I,J) \ast (\text{TB} - \text{TICE}(I,J)) - \text{FW2}(I,J)) + \text{FICE}(I,J)
\]

The snow and ice depth changes are added to all seven levels of the sum and the surface temperature of the ice is saved as the midpoint (TMID2) for the next timestep computations:

\[
\text{TSUM}(I,J) = \text{FICE}(I,J)/\text{NLEVEL} + \text{TSUM}(I,J)
\]

\[
\text{SSUM}(I,J) = \text{HSI}(I,J)/\text{NLEVEL} + \text{SSUM}(I,J)
\]

\[
\text{NZ3} = \text{NLV} - (\text{NLEVEL}+1)/2
\]

if (NZ3 = 0) then \(\text{TMID2}(I,J) = \text{TSFC}(I,J)\).

After values for all seven levels are computed, then the following calculations made:

\[
\text{HSI}(I,J) = \text{SSUM}(I,J)
\]

\[
\text{TSFC}(I,J) = \text{TMID2}(I,J).
\]
This completes all computation for the ice-covered case.

For the open water case, the following are calculated at each x and y thermodynamic field:

\[
\text{TICE}(I,J) = \text{TSFC}(I,J)
\]
\[
\text{Q0}(I,J) = 1.0 \times 10^{-6}/302.0
\]
\[
\text{A1}(I,J) = \text{FSH}(I,J) + \text{FLO}(I,J) + D1 \times \text{UG}(I,J) \times \text{TAIR}(I,J) + D1W \times \text{UG}(I,J) \times \text{QA}(I,J)
\]
\[
\text{B}(I,J) = QS1 \times 6.11 \times \exp(17.2694 \times (\text{TSFC}(I,J) - \text{TMELT})/(\text{TSFC}(I,J) - \text{TMELT} + 237.3))
\]
\[
\text{A2}(I,J) = -D1 \times \text{UG}(I,J) \times \text{TSFC}(I,J) - D1W \times \text{UG}(I,J) \times \text{B}(I,J)
\]
\[
\text{TSUM}(I,J) = \text{Q0}(I,J) \times (-\text{FW2}(I,J) - \text{A1}(I,J) - \text{A2}(I,J) + \text{SNRT} \times 110.0 \times 10^6).
\]

In the equations above, TICE is the temperature of the mixed layer.

The mean ice thickness from all seven levels (TSUM) and the mixed-layer temperature for the open water case or the ice temperature for ice-covered water (TSFC) are returned to the HEAT CSU.

### 3.4 CSC Compute Ocean

#### 3.4.1 CSC Compute Ocean Driver

#### 3.4.1.1 CSU OCEAN

CSU OCEAN serves as the main routine for the CSC Compute Ocean. This routine originated as the main routine in the stand-alone version of the Cox Ocean Model (Cox 1984). Minor modifications were made to allow the original routine to function as a subroutine in the CSCI PIPS2.0.

#### 3.4.1.1.1 CSU OCEAN Design Specification/Constraints – There are no known constraints.

#### 3.4.1.1.2 CSU OCEAN Design

**Input Data Elements:**

- **AREA**
  - surface area of the ocean basin, read in on a restart or computed on a run started from scratch

- **FINS**
  - floating point array of vorticity starting indices read in from the ocean restart file

- **HR**
  - reciprocal depth array read in for model initialization

- **ITT**
  - timestep counter read in on a restart

- **P**
  - stream function computed in previous model execution for the last timestep processed; read in on a restart

- **PB**
  - stream function computed in previous model execution for the second-to-last timestep processed; read in on a restart

- **VOLUME**
  - volume of the ocean basin, read in on a restart or computed on a run started from scratch

**/CORSP/**

- **FCORSP**
  - array of sine of latitude positions for each gridpoint

**/LEVITUS/**

- **SMIX**
  - interpolated monthly salinity data input from the Levitus salinity file

- **TMIX**
  - interpolated monthly temperature data input from Levitus temperature file
Output Data Elements:
/FIELDS/

HR     reciprocal depth array; read in on a restart and computed on a run started from scratch
P      mass transport stream function output to CSU OSTEP to initialize the timestepping
PB     mass transport stream function output to CSU OSTEP to initialize the timestepping
ZTD    array values are initialized to 0

/FULLWD/

AREA   surface area of the ocean basin, read in on a restart or computed on a run started from scratch
EB     sets the type of mixing timestep. If true, a Euler backward timestep is done. If false, a forward step is done.
IEIS    array of ending $I$ indices for island boxes; values are initialized in a DATA statement; = 283, 138, 224, 227
IEZ     array of ending indices for vorticity
ISIS    array of starting $I$ indices for island boxes; values are initialized in a DATA statement; = 272, 133, 220, 225
ISZ     array of starting indices for vorticity
ITT     timestep counter
JEIS    array of ending $J$ indices for island boxes; values are initialized in a DATA statement; = 126, 173, 187, 182
JSIS    array of starting $J$ indices for island boxes; values are initialized in a DATA statement; = 113, 168, 182, 179
KAR     used to enable vectorization
KFLDS   disk unit number for two-dimensional horizontal fields and start and end indices for vorticity; = 12
KONTRL  disk unit number for timestep counter; = 11
LABS(3) array containing disk unit numbers for slabs; = 13, 14, and 15
MXSCAN  maximum number of scans allowed for convergence in CSU CRELAX, initialized in a DATA statement; = 100
NA      initialized in a DATA statement; = 1
NB      initialized to 0
NC      initialized to 0
NDISK   permuting disk number for file containing slab data
NDISKA  permuting disk number for file containing slab data
NDW     initialized to 1,000,000
NMIX    initialized in a DATA statement; = 6
NTSI    initialized in a DATA statement; = 1
NWRITE  initialized to 1,000,000
TTSEC   initialized to 0
VOLUME  volume of the ocean basin, read in on a restart or computed on a run started from scratch

/ONEDIM/

C2DZ    vertical grid spacing times 2; DZ * 2
CS      cosine of U,V point latitudes
CSR     1.0/CS
CST     cosine of T point latitudes
CSTR    1.0/CST
DXT  zonal grid spacing across T boxes (between U,V points)
DXT2R  1.0/(DXT * 2.0)
DXT4R  1.0/(DXT * 4.0)
DXTR  1.0/DXT
DXU  zonal grid spacing across U,V boxes (between T points)
DXU2R  1.0/(DXU * 2.0)
DXU4R  1.0/(DXU * 4.0)
DXUR  1.0/DXU
DYT  meridional grid spacing across T boxes (between U,V points)
DYT2R  1.0/(DYT * 2.0)
DYT4R  1.0/(DYT * 4.0)
DYTR  1.0/DYT
DYU  meridional grid spacing across U,V boxes (between T points)
DYU2R  1.0/(DYU * 2.0)
DYU4R  1.0/(DYU * 4.0)
DYUR  1.0/DYU
DZ(15)  vertical grid spacing in centimeters of U,V,T boxes (between W points):
  DZ(1)  = 3000.0
  DZ(2)  = 4630.0
  DZ(3)  = 6745.0
  DZ(4)  = 9446.0
  DZ(5)  = 12846.0
  DZ(6)  = 17057.0
  DZ(7)  = 22191.0
  DZ(8)  = 28351.0
  DZ(9)  = 35621.0
  DZ(10) = 44057.0
  DZ(11) = 53678.0
  DZ(12) = 64454.0
  DZ(13) = 76300.0
  DZ(14) = 89071.0
  DZ(15) = 102553.0
DZ2R  1.0/(DZ * 2.0)
DZZ  vertical grid spacing across W boxes (between U,V,T points)
DZZ2R  1.0/(DZZ * 2.0)
EEH  upper vertical mixing coefficient of T
EEM  upper vertical mixing coefficient of U,V
FFH  lower vertical mixing coefficient of T
FFM  lower vertical mixing coefficient of U,V
PHI  latitude in radians of the U,V points
PHIT  latitude in radians of the T points
SFU  external mode component of U
SFUB  external mode component of U for the timestep before present
SFV  external mode component of V
SFVB  external mode component of V for the timestep before present
SINE  sine of U,V point latitudes
SINEA  sine of U,V point latitudes in the Earth-oriented spherical coordinates
TINIT  initial values of tracers used only when starting a run from scratch
TNG  tangent of U,V point latitudes
ZDZ  vertical position of bottom of levels
ZDZZ  vertical position of center of levels
ZUN  time change of vertically averaged zonal forcing at north face
ZUS  time change of vertically averaged zonal forcing at south face
ZVN  time change of vertically averaged meridional forcing at north face
ZVS  time change of vertically averaged meridional forcing at south face

/SCALAR/
ACOR  program variable which is assigned the value of ACORF. If equal 0, treat the Coriolis term explicitly. If greater than 0, treat the Coriolis term implicitly with forward component weighted by ACOR
AH  program variable which is assigned the value of AHF; coefficient of horizontal mixing of $T = 1.8 \times 10^7$
AM  program variable which is assigned the value of AMF; coefficient of horizontal mixing of $U,V$
CRIT  program variable which is assigned the value of CRITF; criterion for convergence of relaxation; $= 1.8 \times 10^8$
DTSF  program variable which is assigned the value of DTSFF; length of the timestep on the stream function; $= 180.0$
DTTS  program variable which is assigned the value of DTTSF; length of the timestep on the tracer elements; $= 1800.0$
DTUV  program variable which is assigned the value of DTUVF; length of the timestep on $U,V; = 180.0$
FKPH  program variable which is assigned the value of FKPHF; coefficient of vertical mixing of $T = 1.0$
FKPM  program variable which is assigned the value of FKPMF; coefficient of vertical mixing of $U,V$
GRAV  acceleration due to gravity; $= 980.6$
OMEGA  rate of rotation of the coordinate system; $= PJ/43082$
RADIUS  radian to degree conversion factor; $= 59.29578$
RADIUS  radius of the earth; $= 6370.05$
SOR  program variable which is assigned the value of SORF; coefficient of over-relaxation; $= 1.60$
SWLDEG  latitude in degrees of the southern wall; $= -45.72$

/TSTOP/
CHENG  assigned the Levitus climatological data of temperature and salinity

/WORKSP1/
FKMT  number of vertical levels of ocean at T points
FKMU  number of vertical levels of ocean at U,V points
FMM  FM at row $J - 1$
T  tracer variables, temperature and salinity
TB  tracer variables for the timestep before present
TDIF  diffusion computation array
TEMDA  initialized to 0
TEMPB  initialized to 0
U  zonal component of velocity
UB  zonal component of velocity for timestep before present
UNDER  initialized to 0
V  meridional component of velocity
VB  meridional component of velocity for timestep before present
VUNDER initialized to 0

Parameters:
stored in file ocean.par

GRID 0.5715
GRID2 assigned to DYT and DXT, equal to GRID/INTVL2; = 0.28575
GRID3 assigned to DYT and DXT, equal to GRID/INTVL3; = 0.28575
IMT total number of T grid boxes zonally; = 360
IMTKM IMT * KM; = 5400
IMTM1 IMT – 1; = 359
IMTM2 IMT – 2; = 358
IMTP1 IMT + 1; = 361
IMU total number of U,V grid boxes zonally; = 359
IMUM1 IMU – 1
IMUM2 IMU – 2
INT2 INTVL2 – 1; =1
INT3 INTVL3 – 1; =1
INTVL2 2
INTVL3 2
IRV number of ocean basin grid cells for which river runoff is to be considered; = 88
JMT total number of T grid boxes meridionally; = 360
JMTM1 JMT – 1; = 359
JMTM2 JMT – 2; = 358
JMTP1 JMT + 1; = 361
JSCAN JMTM2
KM total number of vertical levels in ocean basin; = 15
KMM1 KM – 1; = 14
KMP1 KM + 1; = 16
KMP2 KM + 2; = 17
LBC number of arrays of slab data; = 2
LSEG maximum number of sets of start and end indices for islands in ocean basin; = 10
MIDX 100
MIDX2 MIDX + INT2 + INT3 * (MIDX – (NSTLFI1 + 1)); = 200
MIDY 80
MIDYY MIDY + INT2 + INT3 * (MIDY – (NSTLW1 + 1)); = 160
NDICES total number of start and end indices – 2 * LSEG * JMT + 4 * NISLE; = 7216
NGRPDP 181
NGRDPM NGRDP + INT2 + INT3 * NRTLF1 + INT2; = 361
NGRDPY NGRDP + INT2 + INT3 * NUPLW1 + INT2; = 361
NIEVEN 2 * ((NISLE + 1)/2); = 4
NISLE number of islands in the model basin; = 4
NKFLDS number of two-dimensional fields needed on disk unit; 7 + (NDICES/NWDS)
NRTLF1 (NSTRT1–1) – (NRTLF1+1); = 178
NSLAB number of words in one slab; IMT * ((NT + 2) * KM + LBC); = 22320
NSTLF1 0
NSTLW1 0
NSTRT1 180
NSTUP1 180
NSWICH number of words of slab incidental data within each slab that are nonprognostic;
NSWICH words must be switched into correct arrays after being read in on an even timestep; LBC * IMT; = 720
NT
NTMIN2
NUPLW1
NWDS
STLF1
STLW1
STRT1
STUP1

number of tracer elements carried in the model (temperature and salinity); = 2
NT + 1/NT
(NSTUP1–1) – (NSTLW1+1); = 178
IMT * JMT; = 129600
0
0
180.0
180.0

Local Data Files or Databases:
The ocean restart data are read from and written to file fort_<ymmd>.21, where yymmd is the year, month, and day of the run, using the Ocean Direct Access Manager (ODAM) software described in CSC Direct Access Manager (Sec. 3.4.6). Input restart data are read from logical unit 18 and output restart data are written to logical unit 34.

Logic Flow:
Initialize model data:

1. Define parameters through PARAMETER statement. This PARAMETER statement will be repeated in several CSUs.

2. Assign values to program constants. Set x, y, and z dimensions of the grid boxes in centimeters. Assign logical unit numbers. Set radius of the Earth, rate of rotation, acceleration due to gravity and latitude in degrees of the southern wall of the model basin. Set PI and radian to degree conversion factor.

3. Read in file variables:
   a. Read data from the restart file of the ice model. This file contains the ice conditions of the current day including the ice-drift velocity, ice thickness, and ice concentration.
   b. Read ice growth rate in open water, total ice growth rate of the day, and heat above the freezing temperature.
   c. Read Earth-oriented latitudes of each gridpoint.
   d. Read run parameters including control parameters, eddy coefficients, island box corner point indices, initial values of tracers, timestep lengths for T, U, V, and stream function.

4. Compute auxiliary arrays based upon the specified spacing. Methods for computing these auxiliary arrays are straightforward using variable definitions given in the Data Element section of this CSU description.
   a. For Z direction, compute C2DZ, DZ2R, DZZ, ZDZ, DZZ2R, ZDZZ, EEH, FFH, EEM, and FFM. Auxiliary arrays are used for ease in enabling vectorization in other CSUs:
      \[ EEH_k = FKPH/(DZ_k * DZZ_k) \]
      \[ FFH_k = FKPH/(DZ_k * DZZ_{k+1}) \]
      EEM and FFM are defined similarly with FKPM replacing FKPH.
   b. For y direction, compute PHI, PHIT, SUMDY, DYU, DYT, DYTR, DYT2R, DYT4R, DYUR, DYU2R, DYU4R, CST, CS, SINE, CSTR, CSR, and TNG.
   c. For x direction, compute DXU, DXTR, DXT2R, DXT4R, DXUR, DXU2R, and DXU4R.
5. Open the disk datasets including slab data files, control data files, and two-dimensional horizontal field storage files.

Prepare to timestep:
At this point, program execution continues without consideration as to whether this is a model restart or a run from scratch.

1. Read disk data into memory for startup.
   a. Read in timestep counter, total elapsed time, surface area of the model basin, and total volume of the model basin from logical unit number KONTRL.
   b. Read start and end indices from logical unit number KFLDS and convert to integers by storing in ISZ and IEZ array.
   c. Compute permuting disk indicators and read in mass transport stream functions computed for current and previous timesteps. Also read in reciprocal depth array. These values are also stored in the file opened on logical unit number KFLDS.

2. Initialize array variables to 0 including elements of vertical mixing computation arrays, change in vorticity arrays, temporary storage arrays, and diffusion computation arrays.

Timestep the model:
Method: Loop to call CSU OSTEP to compute data for each timestep. Save restart data after every NWRITE calls to CSU OSTEP. After the specified number of timesteps have been taken, close disk units and stop execution.

3.4.2 CSC Ocean Timestep

This CSC controls the processing for each timestep. It prepares the variables needed to compute data for the timestep being processed and drives the main routines for computing tracer and velocity data.

3.4.2.1 CSU OSTEP

CSU OSTEP is the main routine for performing the functions required by CSC Ocean Timestep. It initializes various quantities and bootstraps and manages the I/O for the row-by-row computation of prognostic variables. It also performs analysis procedures on the progressing solution.

3.4.2.1.1 CSU OSTEP Design Specification/Constraints – There are no known constraints.

3.4.2.1.2 CSU OSTEP Design

Input Data Elements:

/FIELDS/
   HR    reciprocal of total depth at U,V points
   PB    mass transport stream function computed for timestep before present

/FULLWD/
   EB
   IITT   timestep counter, incremented by 1 in this CSU
KAR (K) = K; for future vectorization
KFLDS disk unit number of two-dimensional horizontal fields and start and end indices; = 12
KONTRL disk unit number for timestep counter, etc.
LABS disk unit numbers for slabs; = 13, 14, 15
NLAST final timestep to compute on this run of the model
NMIX number of timesteps between mixing timesteps
NERGY number of timesteps between printout of energy data
NTSI number of timesteps between printout of timestep information
TTSEC total elapsed time, incremented by the length of the timestep in this CSU
VOLUME volume of the ocean basin

/ONEDIM/

C2DZ DZ * 2
CS cosine of U,V point latitudes
CSR reciprocal of cosine of U,V point latitudes
CST cosine of T point latitudes
DXT zonal grid spacing across T boxes (between U,V points)
DXT4R 1.0/(DXT * 4.0)
DXU zonal grid spacing across U,V boxes (between T points)
DXU2R 1.0/(DXU * 2.0)
DYU meridional grid spacing across U,V boxes (between T points)
DYUR 1.0/DYU
DYU2R 1.0/(DYU * 2.0)
DZ vertical grid spacing down U,V,T boxes (between W points); vertical layer thickness in centimeters
DZZ vertical grid spacing across W boxes (between U,V,T points); assigned to DZZQ in this CSU
DZZ2R 1.0/(DZZ * 2.0); assigned to DZZ2RQ in this CSU
EEH upper vertical mixing coefficient of T; assigned to EEHQ in this CSU
EEM upper vertical mixing coefficient of U,V; assigned to EEMQ in this CSU
FFH lower vertical mixing coefficient of T; assigned to FFHQ in this CSU
FFM lower vertical mixing coefficient of U,V; assigned to FFMQ in this CSU
SINEA sine of U,V point latitude in Earth-oriented spherical coordinates

/SCALAR/

AH coefficient of horizontal mixing of T
DTSF length of timestep on the mass transport stream function
DTTS length of timestep on T
DTUV length of timestep on U,V
OMEGA rate of rotation of the coordinate system

/TSTOP/

WINDSX zonal wind velocity
WINDSY meridional wind velocity
/WORKSP1/
FKMT  number of vertical levels of ocean at T points for row J
FKMTP number of levels at T points for row J + 1; read in from slab data file
FKMUP number of levels at U points for row J + 1; read in from slab data file
TA    similar to T but after present
TB    tracer data computed for timestep before present
TBP   tracer data for the timestep before present in row J + 1; set to TP on a mixing timestep
TDIF  diffusion computation array
TP    tracer data for row J + 1; read from slab data files
UP    U component of horizontal velocity for row J + 1; read from slab data files
V     meridional component of velocity
VP    V component of horizontal velocity for row J + 1; read from slab data files

Output Data Elements:

/CURNTS/
T0    set equal to T for the last timestep processed
U0    set equal to U for the last timestep processed
V0    set equal to V for the last timestep processed

/FIELDS/
P    mass transport stream function output to disk
PB   mass transport stream function for the timestep before the present

/FULLWD/
BUOY  energy transfer through buoyancy effects
DTABS volume average of absolute change of temperature
EKTOT total kinetic energy normalized by volume
ENGEXT accumulator of rate of change of kinetic energy of external mode; initialized in this CSU
ENGINT accumulator of rate of change of kinetic energy of internal mode; initialized in this CSU
ITT   updated timestep counter
KMT   number of vertical levels of the ocean at T points
KMTP  array containing number of vertical levels of ocean at T points in row J + 1
KMU   number of vertical levels of the ocean at U,V points
KMUP  array containing number of vertical levels of ocean at U,V points in row J + 1
MIX   mixing timestep indicator set in this CSU to indicate whether the current timestep is a mixing timestep
MXP   set to 1 to indicate second pass of a Euler backward timestep
NDISK permutes with NDISKA and NDISKB on logical units 13, 14, and 15 to access slabs at each timestep
NDISKA see NDISK
NDISKB see NDISK
NERGY set to 1 on an energy printout timestep
PLICEX energy change due to implicit effects on external mode
PLICIN energy change due to implicit effects on internal mode
TTDTOT initialized to 0
TTSEC total elapsed time in seconds
TVAR  change of variance of tracers
/ONEDIM/
  SFU  external mode component of U
  SFUB external mode component of U in timestep before present
  SFV  external mode component of V
  SFVB external mode component of V in timestep before present
  ZUS  initialized time change of vertically averaged zonal forcing at south face
  ZVS  initialized time change of vertically averaged meridional forcing at south face

/RVR/
  TRIVER  river temperature

/SCALAR/
  C2DTSF  2 * DTSF; on a mixing timestep equal to DTSF
  C2DTTS  2 * DTTS; on a mixing timestep equal to DTTS
  C2DTUV  2 * DTUV; on a mixing timestep equal to DTUV

/WORKSPI/
  BCON  array of slab incidental data on N + 1 slab
  C2DZQ  set equal to C2DZ for future vectorization
  DXT4RQ set equal to DXT4R for future vectorization
  DXTQ  set equal to DXT for future vectorization
  DXU2RQ set equal to DXU2R for future vectorization
  DXUQ  set equal to DXU for future vectorization
  DZ2RQ  set equal to DZ2R for future vectorization
  DZZ2RQ set equal to DZZ2R for future vectorization
  DZZQ  set equal to DZZ for future vectorization
  EEHQ  set equal to EEH for future vectorization
  EEMQ  set equal to EEM for future vectorization
  FFHQ  set equal to FFH for future vectorization
  FFMQ  set equal to FFM for future vectorization
  FKMUP number of vertical levels of the ocean at U,V points for row J + 1
  FM  masking array for T points
  FMM  masking array for T points in row J - 1
  FMP  masking array for T points in row J + 1
  FVST initialized advective coefficient for south face of T box
  FVSU  advective coefficient for south face of U,V box
  GM  masking array for U,V points
  RHOS initialized values of RHO for the row to the south of the present row
  TA  tracer variables for timestep after present
  TBM  tracer variables for timestep before present for row J - 1
  TBP  tracer variables for timestep before present for row J + 1
  TDIF diffusion computation array
  TM  tracer variables for row J - 1
  UBP  U component of horizontal velocity for timestep before present for row J + 1
  UCLIN array of internal mode component of U at row J + 1
  UP  U component of horizontal velocity for row J + 1
  UUNDER positioned in common to be equivalent with UDIF with K index equal to KM + 1
  VBP  V component of horizontal velocity for timestep before present for row J + 1
  VCLIN array of internal mode component of V at row J + 1
  VUNDER positioned in common to be equivalent with VDIF with K index equal to KM + 1
WSX  zonal component of surface wind stress
WSY  meridional component of surface wind stress
ZUSENG  vertical average of U forcing at south face
ZVSENG  vertical average of V forcing at south face

Parameters:
stored in file ocean.par
The ocean.par parameters are described in Sec. 3.4.1.1.2.

Logic Flow:
Initialize quantities needed for each timestep:

1. Increment timestep counter by 1 and the total elapsed time by timestep increment.

2. Update permuting disk I/O units. At any one time, three files are open for the storage and retrieval of slab and slab incidental data for the $N-1$ (timestep before present), $N$ (present timestep), and $N+1$ (timestep after present) timesteps. For convenience, three permuting disk units are used. For details on the permutation method used by the I/O subsystem refer to Sec. 3.4.6.

3. Prepare timestep length variables C2DTTS, C2DTUV, and C2DTSF. If this is not a mixing timestep, set variables equal to twice the length of the timestep on T, U, and V, and the mass transport stream function. If this is a mixing timestep, set the same variables equal to the length of the timestep on T, U, and V, and the stream function. Also, current mass transport stream function should be saved to the previous mass transport stream function on a mixing timestep.

4. Call CSU STRESSUP to read NOGAPS geostrophic winds file and compute surface wind stress if this is the first timestep.

5. Establish over-dimensioned arrays for vectorization.

6. Call CSU STINIT to load coefficient arrays for subsequent calls to CSU STATE and CSU STATEC.

7. Reset U and V components of the vertical mixing computation arrays UUNDER and VUNDER to 0 for the $KM+1$ boundary.

8. Initialize arrays to contain volume average of absolute change of temperature (DTABS) and change of variance of tracers (TVAR).

9. If this is an energy printout timestep, set indicator flag and prepare necessary variables. Initialize arrays to accumulate rates of change of kinetic energy of internal and external modes (ENGIN, ENGEXT) and vertical average of V forcing at north and south faces of box (ZUSENG, ZVSENG). Also, initialize tracer arrays including arrays to contain integrals (TTDOT), and meridional and northward mass transport of tracers (TMT, TTN).

Bootstrap the row-by-row computation of prognostic variables:
This algorithm bootstraps the row-by-row processing by calculating data for row 1 for the timestep after present.

1. Call CSU OGET to retrieve row 2 tracer data from the disk for the timestep before present and the present timestep.
2. If this is an even timestep, switch slab incidental data into correct arrays. On even timesteps slab incidental data is read into incorrect arrays because of the method used by the direct access subsystem. See description of I/O subsystem in Sec. 3.4.6.

3. Assign arrays containing maximum level indicators at U,V, and T points to integer arrays for use in equations.

4. If this is a mixing timestep, move T and U,V data from the timestep before present to the present timestep for row \( J + 1 \).

5. Initialize arrays for first calls to CSU CLINIC and CSU TRACER including slab arrays, south face of T box advective coefficient arrays, and row \( J \) and \( J - 1 \) land/water masking arrays for T points (FM,FMM). Set all array values to 0.

6. Construct masking array for row 2 T points by checking the contents of the array containing the number of vertical levels at T points. Assign 0 to land points; 1 to water points.

7. Initialize vorticity computation arrays at southern wall (ZUS, ZVS) to 0.

8. Save internal mode velocities for row 2 to arrays UCLIN and VCLIN and compute advective coefficient for south face of row 2 U,V boxes:

\[
FVSU_{i,k} = (VP_{i,k} + V_{i,k}) * FX,
\]

where

\[
FX = DYU2R_2 * CSR_2 * CST_2 * 0.5.
\]

9. Compute external mode velocities for row 2 for the timestep before present and the present timestep from the mass transport stream function results read in CSU OCEAN in preparing to timestep:

\[
SFU_i = ((P_{i+1,j+2} - P_{i,j+1}) + (P_{i,j+2} - P_{i+1,j+1})) * DYU2R_{j+1} * HR_{i,j+1},
\]

\[
SFV_i = ((P_{i+1,j+2} - P_{i,j+1}) - (P_{i,j+2} - P_{i+1,j+1})) * DXU2R_i * HR_{i,j+1} * CSR_{j+1}.
\]

On timestep before present, \( P \) is replaced by \( PB \), and \( SFU \) and \( SFV \) are replaced by \( SFUB \) and \( SFVB \).

10. Add external mode to internal mode for the ocean points in row 2 for present timestep and timestep before present.

11. Accumulate kinetic energy from row 2 every \( NTSI \) timesteps:

\[
EKTOT = EKTOT + (UP_{i,k}^2 + VP_{i,k}^2) * 0.5 * CS_{j+1} * DYU_{j+1} * DZ_k * DXU_i.
\]

12. Call CSU STATE to compute density of row 2.

Perform row-by-row computation of prognostic variables:

Loop for rows 2 through \( JMT - 1 \):

1. Move tracer data for present timestep and timestep before present down one row. Move array elements for row \( J \) data to array elements for row \( J - 1 \). Move row \( J + 1 \) data to row \( J \).

2. Except when processing the last row, read the rest of the \( J + 1 \) slab for the timestep before present and the present timestep.
3. Except when processing row 2 (the first row processed in this algorithm), write newly computed
data from the previous row, row \( J - 1 \). This data was computed in the previous execution of
this row-by-row processing loop.

4. If processing an even timestep, switch maximum level indicators, read in step 2 above into
correct slab. Because of the permuting disk I/O, the maximum level indicators for the \( U,V \)
points are in the \( FKMTP \) array and the maximum level indicators for the \( T \) points are in the
\( FKMUP \) array. The arrays are switched on even timesteps only as is explained in Sec. 3.4.6.

5. Shift maximum level indicators for \( T \) and \( U,V \) points from row \( J + 1 \) to row \( J \) and set \( J + 1 \)
floating point values to integer. Maximum level indicators should be in arrays \( KMT \) and \( KMU \)
for row \( J \) and \( KMTP \) and \( KMUP \) for row \( J + 1 \).

6. If on a mixing timestep, set \( \tau - 1 \) slab data equal to \( \tau \) level slab data. Slab data includes
tracer data and \( U \) and \( V \) components of horizontal velocity.

7. Shift masks down one row and compute new masks. The masking arrays are \( FM \) for row \( J \),
\( FMM \) for row \( J - 1 \), \( FMP \) for row \( J + 1 \). Set new values for row \( J \) into \( FMP \) based on contents of
\( KMTP \) arrays. Also set \( GM \) masking array based on contents of \( KMU \).

8. Call CSU CLINIC to update the internal mode and vorticity driving function for row \( J \).

9. Call CSU TRACER to update the tracer quantities for row \( J \).

10. Print the progressing solution at every fourth row on energy timestep. Call CSU MATRIX to
print arrays of temperature and salinity tracers and \( U,V \), and \( W \) components of velocity.

11. If not processing the northernmost row, compute the northward transport of each tracer quantity,
as well as the zonally integrated meridional mass transport.

   a. For temperature and salinity tracers, move data computed for the north of the previous row
to the south of the current row.

   b. Sum all tracer data for row 2, masking out land points with the \( FM \) array so that land grid
cells do not contribute to the sum. Average tracer data by dividing by the total ocean area
of row 2:

\[
TBRS_{k,m} = TBRS_{k,m} + T_{i,k,m} \cdot FM_{i,k} \cdot DXT_i,
\]

\[
TBRS_{k,m} = TBRS_{k,m} / TOTDX,
\]

where \( TOTDX \) is the sum of the zonal grid cells for all ocean gridpoints in row 2. Row 2
land points are not included in the sum.

   c. Compute \( TBRN \) similarly using masking array and data computed for row \( J + 1 \) in timestep
before present.

\[
TBRN_{k,m} = TBRN_{k,m} + TP_{i,k,m} \cdot FMP_{i,k} \cdot DXT_i,
\]

\[
TBRN_{k,m} = TBRN_{k,m} / TOTDX.
\]

d. Sum the horizontal velocity zonally.

\[
VBR_k = VBR_k + V_{i,k} \cdot DXU_i \cdot CS_j.
\]

e. Compute the meridional mass transport by successively adding \( VBR \) contributions at each
depth level.
For depth level 1:
\[ TMT_{j,1} = VBR_1 \times DZ_1. \]

For all other depth levels:
\[ TMT_{j,k} = TMT_{j,k-1} + VBR_k \times DZ_k. \]

f. Compute \( TTN \), the northward transport of the tracers:
\[ TTN_{1,i,m} = TTN_{1,i,m} + VBR_k \times (TBRN_{k,m} + TBRS_{k,m}) \times 0.5 \times DZ_k \]

g. Compute zonal/vertical averages of tracers and horizontal velocity:
\[ VBRZ = VBRZ + (V_{i,k} \times DXU_i + V_{i-1,k} \times DXU_{i-1}) \times DZ_k, \]
\[ VBRZ = VBRZ / TOTDZ, \]
where \( TOTDZ \) is the total vertical grid spacing,
\[ TBRZ = TBRZ + T_{i,k,m} + TP_{i,k,m} \times DZ_k. \]

h. If processing the last timestep, write data to the file connected to logical unit 10. This is the file containing the ocean conditions of the current day.

12. Move slab incidental data into the correct slab for writing using the temporary array \( BCON \). On an even timestep, set \( BCON \) values equal to \( FKMT \) values. Otherwise, set \( BCON \) values equal to \( FKMU \) values. These data are written out at the beginning of the next execution of this loop (see step 3 above of this algorithm).

End loop for row-by-row computation.

Write out computed data:

1. Print one line of timestep information every \( NTSI \) timesteps, including timestep number, total kinetic energy, volume average of absolute change in temperature and salinity, and number of scans to convergence in CSU CRELAX.

2. Complete and print the on-line integrals on energy timesteps.
   a. Normalize previously computed integrals by dividing by volume. This includes total change of kinetic energy on internal and external modes, tracer integrals, and total change of variance on tracers.
   b. Compute residuals for energy change due to implicit effects on internal and external modes and tracer integrals.


4. Initiate write-out of newly computed data from the final row.

Prepare for the next timestep:

1. Call CSU CRELAX to solve for the new mass transport stream function.

2. If this is the end of the first pass of a Euler backward timestep, set the input disk units so that the proper levels are fetched on the next pass. The output for the second pass will be placed on the \( NDISKA \) unit. Return to the top of CSU OSTEP to do the second pass if this is the first pass of a Euler backward timestep.
3. For purposes of recovering from the disk after an abnormal stop, bring otherwise inactive disk units up-to-date by writing the new mass transport stream function and the total number of timesteps that were completed.

4. If this is an energy timestep, print the new stream function using CSU MATRIX.

### 3.4.2.2 CSC Compute Wind/Ice Stress

CSC Compute Wind/Ice Stress requires the computation of the wind/ice stress using a wind drag coefficient which varies as a function of ice thickness.

#### 3.4.2.2.1 CSU STRESSUP

CSU STRESSUP meets all of the requirements specified for the CSC Compute Wind/Ice Stress.

#### 3.4.2.2.1.1 CSU STRESSUP Design Specification/Constraints. Compute wind/ice stress using a wind drag coefficient which varies as a function of ice thickness according to the law of the wall.

#### 3.4.2.2.1.2 CSU STRESSUP Design

**Input Data Elements:**

```
/COX2/
FW1    heat above the freezing temperature
GICE   ice growth rate computed by the ice model
SHICE  growth rate of ice thickness

/CURNTS/
T0     set equal to T for the last timestep processed
U0     set equal to U for the last timestep processed
V0     set equal to V for the last timestep processed

/FULLWD/
EKTOT  total kinetic energy normalized by volume
ENGEXT accumulators of rates of change of kinetic energy of external mode
ENGINT accumulators of rates of change of kinetic energy of internal mode
ITT    timestep counter; total number of timesteps completed
KAR    KAR(K) = K; used to enable vectorization
KMU    number of vertical levels of ocean at U,V points
KMUP   number of vertical levels of ocean at U,V points for row J + 1
MXP    if = 1, second pass of Euler backward timestep
NERGY  if = 1, indicates energy printout timestep
NTSI   number of timesteps between prints of a single line of timestep information

/RFOR2/
GAIRX  x component of the geostrophic wind
GARY   y component of the geostrophic wind

/RSTRT/
AREA1  fraction of grid cell covered with ice
HEFF   mean ice thickness per grid cell
```
UICE x component of ice drift
VICE y component of ice drift

/TSTEP/
IDTG date-time group in the form YYMMDDHH
ITSTEP number of timesteps for run
MDY day calculated from inputted data
MHR hour calculated from inputted data
MM month calculated from inputted data
MYR year calculated from inputted data

/TSTOP/
WINDSX zonal wind velocity
WINDSY meridional wind velocity

Output Data Elements:

Parameters:
stored in file ocean.par
The ocean.par parameters are described in Sec. 3.4.1.1.2.

Local Data Elements:
CDWI drag coefficient between water and ice as a function of ice thickness according to the law of the wall
CONSTK the von Karman constant, 0.41
CONSTZ minimum sea ice roughness, 0.01 m
HMIN minimum ice thickness
I index counter
J index counter
TX temporary array used in computing wind stress
TY temporary array used in computing wind stress

Logic Flow:
1. For all surface grid cells except the last row and column, compute wind/ice stress using compute wind drag coefficient, CDWI.

Compute wind drag coefficient:

\[ CDWI = 1.0/(DLOG(HMIN/CONSTZ)/CONSTK) * * 2, \]

where

\[ HMIN = \text{maximum of the ice thickness for grid cell } I,J \text{ and } 0.015, \]
\[ CONSTZ = 0.01, \text{ and } \]
\[ CONSTK = 0.41. \]
Compute:

\[ TX_{i,j} = WINDSX_{i,j} \times |WINDSX_{i,j}| \times 0.0013 \times 0.0008 \times 10000.0 \times (1.0 - AICE_{i,j,2}) \]
\[ \quad + (UICE_{i,j,2} \times 100.0 - U1_{i,j}) \times |UICE_{i,j,2}| \times 100.0 - U1_{i,j} \times CDWI \times AICE_{i,j,2} \]

\[ TY_{i,j} = WINDSY_{i,j} \times |WINDSY_{i,j}| \times 0.0013 \times 0.0008 \times 10000.0 \times (1.0 - AICE_{i,j,2}) \]
\[ \quad + (VICE_{i,j,2} \times 100.0 - V1_{i,j}) \times |VICE_{i,j,2}| \times 100.0 - V1_{i,j} \times CDWI \times AICE_{i,j,2} \]

2. Set \( WINDSX \) and \( WINDSY \) for all surface grid cells. Last row and column values will be 0. \( WINDSX_{i,j} = TX_{i,j}/10.0 \) and \( WINDSY_{i,j} = TY_{i,j}/10.0 \).

3.4.2.3 CSC Load Normalization Constants

This sublevel CSC requires a CSU to load normalization constants into arrays for use in computing densities.

3.4.2.3.1 CSU STINIT - CSU STINIT loads normalization constants into arrays for subsequent calls to CSU STATE and CSU STATEC.

3.4.2.3.1.1 CSU STINIT design specification/constraints. Overdimension arrays are provided to permit future vectorization.

3.4.2.3.1.2 CSU STINIT design.

**Output Data Elements:**

/WORKSP/

- **CQ**: coefficients of normalized temperatures and salinities used in CSU STATE
- **TOQ**: normalizing temperatures used in CSU STATE
- **SOQ**: normalizing salinities used in CSU STATE
- **CIQ**: coefficients of normalized temperatures and salinities used in CSU STATEC
- **TOIQ**: normalizing temperatures used in CSU STATEC
- **SOIQ**: normalizing salinities used in CSU STATEC

**Local Data Elements:**

- **C**: coefficients of equation of state assigned in a DATA statement
- **SO**: normalizing salinities
- **TO**: normalizing temperatures

**Parameters:**

stored in file ocean.par

The ocean.par parameters are described in Sec. 3.4.1.1.2.

**Logic Flow:**

Load coefficients of equation of state into overdimensioned array for use in CSU STATE.

Load normalizing temperatures and salinities into overdimensioned arrays for use in CSU STATE.

Load coefficients of equation of state for use in CSU STATEC. Determine the reference level indicator so that the coefficients for the two levels being compared are equal. Two passes are required to set two sets of coefficients. In the first pass, values for the first set of elements are assigned, where levels 1 and 2 are assigned level 2 coefficients, levels 3 and 4 are assigned level 4 coefficients, and so on. In the second pass, values for the second set of elements are
assigned, where levels 1 and 2 are assigned level 1 coefficients, levels 3 and 4 are assigned level 3 coefficients, and so on.

Load normalizing temperatures and salinities into overdimensioned arrays for use in CSU STATEC using the method described above.

3.4.2.4 CSC Compute Densities

CSC Compute Densities requires the computation of normalized densities.

3.4.2.4.1 CSU STATE and CSU STATEC – CSU STATE and CSU SINIT are used to compute normalized densities using a third-order fit to the Knudsen formula. CSU STATE is called by CSU CLINIC and by CSU TIMESTEP in the bootstrap procedure. CSU STATEC is called by CSU TRACER.

3.4.2.4.1.1 CSU STATE and CSU STATEC design specification/constraints. This subroutine contains two entry points for computing density.

3.4.2.4.1.2 CSU STATE and CSU STATEC design.

Input/Output Data Elements:
- RHO: the returned row of normalized densities
- TX: the input row of temperatures
- SQ: one row of workspace provided by the calling routine
- TQ: one row of workspace provided by the calling routine
- SX: the input row of salinities in units (ppt–35)/1000
- IND: for CSU STATEC only:
  - if IND = 1, compare levels 1 to 2, 3 to 4, etc.;
  - if IND = 2, compare levels 2 to 3, 4 to 5, etc.
  - in either case, use coefficients for the lower of the two levels

/WORKSP/
- SOQ: normalizing salinities
- TOQ: normalizing temperatures

Parameters:
stored in file ocean.par
The ocean.par parameters are described in Sec. 3.4.1.1.2.

Logic Flow:
Compute normalized densities by using a third-order polynomial fit to the Knudsen formula

1. Subtract normalizing constants from input temperature and salinity.
2. Compute polynomial approximation of Knudsen density:

\[
RHO_{i,k} = (CQ_{i,k,1} + (CQ_{i,k,4} + CQ_{i,k,7} \cdot SQ_{i,k}) \cdot SQ_{i,k} + (CQ_{i,k,3} + CQ_{i,k,8} \cdot SQ_{i,k} + CQ_{i,k,6} \cdot TQ_{i,k}) \cdot TQ_{i,k}) + (CQ_{i,k,2} + CQ_{i,k,5} + CQ_{i,k,9} \cdot SQ_{i,k}) \cdot SQ_{i,k}^2
\]

For entry STATEC, array CIQ replaces array CQ and the index of the fourth dimension of CIQ is equal to IND.
3.4.2.5 CSC Print Matrix

CSC Print Matrix requires a subroutine to print elements of a two-dimensional array.

3.4.2.5.1 CSU MATRIX – CSU MATRIX is called by CSU OSTEP to print elements of a two-dimensional matrix on specified timesteps.

3.4.2.5.1.1 CSU MATRIX design specification/constraints.

3.4.2.5.1.2 CSU MATRIX design.

Input Data Elements:

- **ARRAY** the array to be printed
- **IRDIM** the first dimension of the array
- **ISTRT** the first element of the first dimension to be printed
- **IM** the last element of the first dimension to be printed
- **JM** the last element of the second dimension to be printed – the rows are printed in descending order (if \( JM = 0, KK \) is used)
- **KK** the last element of the second dimension to be printed – the rows are printed in ascending order (if \( KM = 0, JM \) is used)
- **SCALE** a scaling factor by which array is divided before printing. If scale = 0, no scaling is done and 10 columns are printed across in E format. If scale > 0, scaling is performed and 20 columns are printed across in F format.

Local Data Elements:

- **IDIF** number of elements to be printed across a line; the difference between IE and IS plus 1
- **IE** ending element number when printing a line of values
- **IS** starting element number when printing a line of values
- **JORK** do loop index for second dimension
- **JMORKM** ending element of second dimension, equals \( JM + KK \)
- **L** the row number printed on a line
- **NUM** array for temporary storage, unscaled ARRAY elements printed across a line
- **PLINE** array for temporary storage, ARRAY elements multiplied by SCALER
- **SCALER** reciprocal of scale factor, used to multiple array elements

Parameters:

stored in file ocean.par

The ocean.par parameters are described in Sec. 3.4.1.1.2.

Logic Flow:

1. If SCALE = 0, print array elements in groups of 10. Print column numbers across the top. Print row number followed by 10 array elements using E13.5 format. Repeat until all desired values are printed.

2. Otherwise, print array elements in groups of 20. Print column numbers across the top. Print row number followed by 20 array elements using F6.2 format. Repeat until all desired values are printed.

3.4.3 CSC Compute Internal Mode and Vorticity Driving Function

This CSC requires computation of the internal mode component of the U and V velocities and the vorticity driving function for use in determining the external mode.
### 3.4.3.1 CSU CLINIC

CSU CLINIC meets the requirements of CSC Compute Internal Mode and Vorticity Driving Function.

#### 3.4.3.1.1 CSU CLINIC Design Specification/Constraints – There are no known constraints.

#### 3.4.3.1.2 CSU CLINIC Design

**Input Data Elements:**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKTOT</td>
<td>total kinetic energy normalized by volume</td>
</tr>
<tr>
<td>ENGEXT</td>
<td>accumulators of rates of change of kinetic energy of external mode</td>
</tr>
<tr>
<td>ENGINF</td>
<td>accumulators of rates of change of kinetic energy of internal mode</td>
</tr>
<tr>
<td>ITT</td>
<td>timestep counter; total number of timesteps completed</td>
</tr>
<tr>
<td>KAR</td>
<td>KAR(K) = K; used to enable vectorization</td>
</tr>
<tr>
<td>KMU</td>
<td>number of vertical levels of ocean at U,V points</td>
</tr>
<tr>
<td>KMUP</td>
<td>number of vertical levels of ocean at U,V points for row J + 1</td>
</tr>
<tr>
<td>MXP</td>
<td>if = 1, second pass of Euler backward timestep</td>
</tr>
<tr>
<td>NERGY</td>
<td>if = 1, indicates energy printout timestep</td>
</tr>
<tr>
<td>NTSI</td>
<td>number of timesteps between prints of a single line of timestep information</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>cosines of U,V point latitudes</td>
</tr>
<tr>
<td>CSR</td>
<td>reciprocal of the cosines of the U,V point latitudes</td>
</tr>
<tr>
<td>CST</td>
<td>cosines of T point latitudes</td>
</tr>
<tr>
<td>CSTR</td>
<td>reciprocal of the cosines of T point latitudes</td>
</tr>
<tr>
<td>DXT</td>
<td>zonal grid spacing across T boxes</td>
</tr>
<tr>
<td>DXT2R</td>
<td>1.0/(DXT * 2.0)</td>
</tr>
<tr>
<td>DXT4R</td>
<td>1.0/(DXT * 4.0)</td>
</tr>
<tr>
<td>DXTR</td>
<td>1.0/DXT</td>
</tr>
<tr>
<td>DXU</td>
<td>zonal grid spacing across U,V boxes</td>
</tr>
<tr>
<td>DXU2R</td>
<td>1.0/(DXU * 2.0)</td>
</tr>
<tr>
<td>DXUR</td>
<td>1.0/DXU</td>
</tr>
<tr>
<td>DYT</td>
<td>meridional grid spacing across T boxes</td>
</tr>
<tr>
<td>DYT2R</td>
<td>1.0/(DTY * 2.0)</td>
</tr>
<tr>
<td>DYTTR</td>
<td>1.0/DYT</td>
</tr>
<tr>
<td>DYU</td>
<td>meridional grid spacing across U,V boxes</td>
</tr>
<tr>
<td>DYU2R</td>
<td>1.0/(DYU * 2.0)</td>
</tr>
<tr>
<td>DYU4R</td>
<td>1.0/(DYU * 4.0)</td>
</tr>
<tr>
<td>DYUR</td>
<td>1.0/DYU</td>
</tr>
<tr>
<td>DZ</td>
<td>vertical grid spacing down U,V,T boxes; vertical layer thickness in centimeter</td>
</tr>
<tr>
<td>DZZ</td>
<td>vertical grid spacing down W boxes</td>
</tr>
<tr>
<td>EEM</td>
<td>upper vertical mixing coefficient of U,V</td>
</tr>
<tr>
<td>FFM</td>
<td>lower vertical mixing coefficient of U,V</td>
</tr>
<tr>
<td>SINE</td>
<td>sine of U,V point latitude</td>
</tr>
<tr>
<td>SINEA</td>
<td>sine of U,V point latitude in Earth-oriented coordinates; used to compute Coriolis force</td>
</tr>
<tr>
<td>TNG</td>
<td>tangent of U,V point latitude</td>
</tr>
<tr>
<td>ZUS</td>
<td>time change of vertically averaged zonal forcing at south face</td>
</tr>
<tr>
<td>ZVS</td>
<td>time change of vertically averaged meridional forcing at south face</td>
</tr>
</tbody>
</table>
/SCALAR/
ACOR if >0, treat the Coriolis term implicitly with forward component weighted by ACOR, past component by 1−ACOR
AM coefficient of horizontal mixing of U,V
C2DTSF DTSF * 2.0
C2DTUV DTUV * 2.0
FKPM coefficient of vertical mixing of U,V
GRAV acceleration due to gravity; = 980.6 cm/s²
OMEGA rate of rotation of the coordinate system
RADIUS radius of the Earth

/WORKSPI/
C2DZQ DZ * 2.0; for future vectorization
DXT4RQ DXT4R (1.0/(DXT * 4.0)); for future vectorization
DXU2RQ DXU2R (1.0/(DXU * 2.0)); for future vectorization
DXUQ DXU; for future vectorization
DZZQ DZZ; for future vectorization
EEMQ EEM ; for future vectorization
FFMQ FFM; for future vectorization
FMM lower vertical mixing coefficient of U,V
FVSU advective coefficient for south face of U,V box
GM masking array for U,V points
RHON density with a space and time invariant subtracted, for the row to the north of the current row
RHOS density with a space and time invariant subtracted, for the row to the south of the current row
TDIF diffusion computation array
TP tracer variables computed for row J + 1
U zonal component of horizontal velocity
UB zonal component of horizontal velocity for the timestep before the present timestep
UBM UB for row J − 1
UBP UB for row J + 1
UCLIN array of internal mode component of U at row J + 1
UM zonal component of horizontal velocity for row J − 1
UP U for row J + 1
V meridional component of horizontal velocity
VB meridional component of horizontal velocity for the timestep before the present timestep
VCLIN array of internal mode component of V at row J + 1
VM V for row J − 1
VP V for row J + 1
WSX zonal component of surface wind stress
WSY meridional component of surface wind stress
ZUSENG vertical average of U forcing at south face
ZVSENG vertical average of V forcing at south face

Output Data Elements:

/FULLWD/
EKTOT total kinetic energy normalized by volume
ENGEEXT accumulators of rates of change of kinetic energy on external mode
ENGINT accumulators of rates of change of kinetic energy on internal mode
/ONEDIM/
SFU    external mode component of U
SFUB   external mode component of U for timestep before present
SFV    external mode component of V
SFVB   external mode component of V for timestep before present
ZUN    time change of vertically averaged zonal forcing at north face
ZUS    time change of vertically averaged zonal forcing at south face
ZVN    time change of vertically averaged meridional forcing at north face

/WORKSPI/
FUW    zonal component of advective coefficient for west face of U,V box
FVN    meridional component of advective coefficient for north face of U,V box
FVSU   advective coefficient for south face of U,V box
TDIF   diffusion computation array
TEMPA  utility array used as temporary storage
TEMPB  utility array used as temporary storage
TP     similar to T but in row J + 1
UA     zonal component of velocity for timestep after present

Parameters:
stored in file ocean.par

The ocean.par parameters are described in Sec. 3.4.1.1.2.

Logic Flow:
Prepare arrays for the computation of internal modes:
Several arrays must be calculated to prepare for the computation of the internal modes of horizontal velocity. These include:
• the advective coefficients for the west (FUW) and north (FVN) faces of the U,V boxes,
• the zonal and meridional components of external mode,
• horizontal velocities for row J + 1 for tau and tau – 1 time levels,
• the density for row J + 1,
• the vertical velocity for U,V columns,
• hydrostatic pressure gradients, and
• boundary conditions for the computation of the vertical diffusion of momentum.

1. Find advective coefficients for west and north faces of U,V box using data computed from the timestep before present. These variables were read in CSU OSTEP:
   a. Calculate external mode component of U (SFU) at west face of U,V box and external mode component of V (SFV) at north face of U,V box. Use mass transport stream function read in CSU OSTEP:
      $SFU_i = -(P_{i,j+1} - P_{i,j}) * DYUR_j * HR_{min},$
      where $HR_{min}$ is the minimum of $HR_{i-1,j}$ and $HR_{i,j}$, and
      $SFV_i = (P_{i+1,j+1} - P_{i,j+1}) * DXUR_i * HR_{min} * CSTR_{j+1},$
      where $HR_{min}$ is the minimum of $HR_{i,j+1}$ and $HR_{i,j}.$
b. Calculate internal mode component of $U$ at west face of $U,V$ box and internal mode component of $V$ at north face of $U,V$ box:

\[ FUW_{i,k} = (UCLIN_{i,k} + UCLIN_{i-1,k}) \times 0.5 \]

\[ FVN_{i,k} = (VP_{i,k} + VCLIN_{i,k}) \times 0.5. \]

c. Add external modes to internal modes and add grid weight factor to obtain advective coefficients where

\[ CSR_j = 1/cos(lat_{i,j}) \]

is the grid weight factor for the west face of $U,V$ box and

\[ FX = DUYU_2R_{j+1} \times CSR_j \times CST_{j+1} \]

is the grid weight factor for the north face of $U,V$ box.

2. Compute external mode velocities for row $J+1$ for present timestep and timestep before present. External mode velocities are computed from the mass transport stream function, $P$:

\[ SFU = -((P_{i+1,j+2} - P_{i,j+1}) + (P_{i,j+2} - P_{i+1,j+1})) \times DUYU_2R_{j+1} \times HR_{i,j+1} \]

\[ SFV = ((P_{i+1,j+2} - P_{i,j+1}) - (P_{i,j+2} - P_{i+1,j+1})) \times DXU_2R_{i+1} \times HR_{i+1,j+1} \times CSR_{j+1}. \]

On timestep before present, $PB$ replaces $P$ and results are stored in $SFUB$ and $SFVB$.

3. Add external mode to internal mode for all ocean points in row $J+1$ for present timestep and timestep before present.

4. Accumulate kinetic energy from row $J+1$ every NTSI timesteps.

Compute contribution to total kinetic energy:

\[ UENG_{i,k} = (FX \times (UP_{i,k}^2 + VP_{i,k}^2)) \times C2DZQ_{i,k} \times DXUQ_{i,k}. \]

where \[ FX = 0.25 \times CS_{j+1} \times DUY_{j+1}. \]

Add $UENG_{i,k}$ to total kinetic energy stored in $EKTOT$.

5. Call CSU STATE to compute density of row $J+1$. Store computed results in RHON.


a. Set vertical velocity at the surface to 0 (rigid-lid), and set vertical velocity at maximum level to 0. The rigid-lid assumption of 0 vertical motion at the surface filters out external gravity waves that would otherwise limit the timestep of the numerical integration.

b. Compute change of vertical velocity $W$ between levels

\[ W_{i,k+1} = C2DZQ_{i,k} \times ((FUW_{i+1,k} - FUW_{i,k}) \times DXU2RQ_{i,k} + FVN_{i,k} - FVSU_{i,k}). \]

c. Integrate downward from the surface by adding successive vertical velocities:

\[ W_{i,k+1} = W_{i,k} + W_{i,k+1}. \]
7. Compute hydrostatic pressure gradient.
   
a. Compute it at the first level:
   \[
   UDIF_{i,1} = RHON_{i+1,1} - RHOS_{i,1} \\
   VDIF_{i,1} = RHON_{i,1} - RHOS_{i+1,1} \\
   DPDX_{i,1} = ((UDIF_{i,1} - VDIF_{i,1}) * FXA) * DXU2R_i \\
   DPDY_{i,1} = (UDIF_{i,1} + VDIF_{i,1}) * FXB, \\
   \]
   where \( FXA = GRAV * DZZ_1 * CSR_j \) and \( FXB = GRAV * DZZ_1 * DYU2R_j \).

b. Compute the change in pressure gradient between levels:
   \[
   DPDX_{i,k} = RHON_{i,k-1} + RHON_{i,k} \\
   DPDY_{i,k} = RHOS_{i,k-1} + RHOS_{i,k} \\
   UDIF_{i,k} = DPDX_{i,k+1} - DPDY_{i,k} \\
   VDIF_{i,k} = DPDX_{i,k} - DPDY_{i,k+1} \\
   DPDX_{i,k} = (FXA * (UDIF_{i,k} - VDIF_{i,k})) * DZZQ_{i,k} * DXU2RQ_{i,k} \\
   DPDY_{i,k} = (FXB * (UDIF_{i,k} + VDIF_{i,k})) * DZZQ_{i,k}. \\
   \]

c. Integrate downward from the first level:
   \[
   DPDX_{i,k+1} = DPDX_{i,k} + DPDX_{i,k+1} \\
   DPDY_{i,k+1} = DPDY_{i,k} + DPDY_{i,k+1}. \\
   \]

8. Set the boundary conditions for the computation of vertical diffusion of momentum.
   
a. Transfer interior points into diffusion computation arrays. Store \( UB \) and \( VB \) elements into \( UDIF \) and \( VDIF \) arrays.

b. Set surface \((K=0)\) elements of diffusion computation arrays to reflect wind stress:
   \[
   UOVER_i = UB_{i,1} + WSX_i * FX, \\
   VOVER_i = VB_{i,1} + WSY_i * FX, \\
   \]
   where \( FX = DZZ_1/FKPM \).

c. Set first land level in each column to reflect bottom condition. Assume a 10° turning angle at the bottom boundary:
   \[
   UDIF_{i,kz+1} = UB_{i,kz} - FXB * (UB_{i,kz} * 0.98481 - VB_{i,kz} * 0.17365) \\
   VDIF_{i,kz+1} = VB_{i,kz} - FXB * (UB_{i,kz} * 0.17365 + VB_{i,kz} * 0.98481). \\
   \]

**Begin computation of the internal modes.**

Method: First, the total advection of momentum is calculated from the flux through the west face of the U,V box, the meridional and zonal flux divergence, the flux through the top of the U,V box, and the vertical flux divergence. Next, horizontal diffusion of momentum is evaluated for a \( \tau - 1 \) timestep and vertical diffusion of momentum is determined from gradients at top of U,V box and from upper and lower vertical mixing coefficients of U and V. Coriolis force is computed at \( \tau \) timestep for explicit treatment and at \( \tau - 1 \) timestep for implicit treatment. Adding these
elements to the computed hydrostatic pressure gradient yields the time rate of change of velocity. New velocities are computed by multiplying the time rate of change by twice the length of the timestep on U,V and adding this product to the U and V components of velocity computed for the timestep before present. These velocities are corrected by finding the incorrect vertical means and subtracting them out.

1. Compute total advection of momentum.
   a. Compute flux through west face of U,V box:

   \[ \text{TEMPA}_{i,k} = F\text{UW}_{i,k} * (U_{i-1,k} + U_{i,k}) \]
   \[ \text{TEMPB}_{i,k} = F\text{UW}_{i,k} * (V_{i-1,k} + V_{i,k}) \]

   b. Compute zonal flux divergence:

   \[ UA_{i,k} = (\text{TEMPA}_{i,k} - \text{TEMPA}_{i+1,k} * DXU2RQ_{i,k}) \]
   \[ VA_{i,k} = (\text{TEMPB}_{i,k} - \text{TEMPB}_{i+1,k} * DXU2RQ_{i,k}) \]

   c. Add in meridional flux divergence:

   \[ UA_{i,k} = UA_{i,k} - F\text{VN}_{i,k} * (UP_{i,k} + U_{i,k}) + F\text{VSU}_{i,k} * (U_{i,k} + UM_{i,k}) \]
   \[ VA_{i,k} = VA_{i,k} - F\text{VN}_{i,k} * (VP_{i,k} + V_{i,k}) + F\text{VSU}_{i,k} * (V_{i,k} + VM_{i,k}) \]

   d. Compute flux through top of U,V box:

   \[ \text{TEMPA}_{i,k} = W_{i,k} * (U_{i,k-1} + U_{i,k}) \]
   \[ \text{TEMPB}_{i,k} = W_{i,k} * (V_{i,k-1} + V_{i,k}) \]

   e. Add in vertical flux divergence:

   \[ UA_{i,k} = UA_{i,k} + \text{TEMPA}_{i,k+1} - \text{TEMPA}_{i,k} * DZ2RQ_{i,k} \]
   \[ VA_{i,k} = VA_{i,k} + \text{TEMPB}_{i,k+1} - \text{TEMPB}_{i,k} * DZ2RQ_{i,k} \]

2. Add in horizontal diffusion of momentum (evaluated at \( \tau - 1 \) timestep).

   a. Compute coefficients dependent only on latitude:

   \[ BB_{UJ} = 8.0 * AM * CSR_j^2 \]
   \[ CC_{UJ} = AM * CST_j * DYTR_{j+1} * DYUR_j * CSR_j \]
   \[ DD_{UJ} = AM * CST_j * DYTR_j * DYUR_j * CSR_j \]
   \[ GG_{UJ} = AM * (1.0 - TNG_j^2)/RADIUS^2 \]
   \[ HH_{UJ} = 2.0 * AM * SINE_j/(RADIUS * CS_j^2) \]

   b. Compute gradients at west face of U,V box:

   \[ \text{TEMPA}_{i,k} = DXT4RQ_{i,k} * (UB_{i,k} - UB_{i-1,k}) \]
   \[ \text{TEMPB}_{i,k} = DXT4RQ_{i,k} * (VB_{i,k} - VB_{i-1,k}) \]

   c. Add in final contribution from horizontal diffusion of momentum:

   \[ UA_{i,k} = UA_{i,k} + BB_{UJ} * (DXU2RQ_{i,k} * (\text{TEMPA}_{i+1,k} - \text{TEMPA}_{i,k}) + 
   CC_{UJ} * (UBP_{i,k} - UB_{i,k}) + DD_{UJ} * (UBM_{i,k} - UB_{i,k}) + GG_{UJ} * 
   UB_{i,k} - HH_{UJ} * DXU2RQ_{i,k} * (VB_{i+1,k} - VB_{i-1,k}) \]
\[ VA_{i,k} = VA_{i,k} + BBUJ \times (DXU2RQ_{i,k} \times (TEMPB_{i+1,k} - TEMPB_{i,k}) + CCUJ \times (VB_{i,k} - VB_{i-1,k}) + DDUJ \times (VBM_{i,k} - VB_{i,k}) + GGUJ \times VB_{i,k} + HHUJ \times DXU2RQ_{i,k} \times (UB_{i+1,k} - UB_{i-1,k}). \]

3. Add in vertical diffusion of momentum.
   a. Compute gradients at top of U,V box:
      \[ TEMP_{A_{i,k}} = UDIF_{i,k-1} - UDIF_{i,k} \]
      \[ TEMP_{B_{i,k}} = VDIF_{i,k-1} - VDIF_{i,k}. \]
   b. Add in final contribution from vertical diffusion of momentum:
      \[ UA_{i,k} = UA_{i,k} + EEMQ_{i,k} \times TEMPA_{i,k} - FFMQ_{i,k} \times TEMPA_{i,k+1} \]
      \[ VA_{i,k} = VA_{i,k} + EEMQ_{i,k} \times TEMPB_{i,k} - FFMQ_{i,k} \times TEMPB_{i,k+1}. \]

4. Add in Coriolis force. Evaluate at tau timestep for explicit treatment and at tau - 1 timestep for implicit treatment with remainder of term to be added later.
   Explicit treatment of Coriolis force:
   \[ UA_{i,k} = UA_{i,k} + FX \times V_{i,k} \]
   \[ VA_{i,k} = VA_{i,k} - FX \times U_{i,k}. \]
   Implicit treatment of Coriolis force:
   \[ UA_{i,k} = UA_{i,k} - FX \times VB_{i,k} \]
   \[ VA_{i,k} = VA_{i,k} - FX \times UB_{i,k}. \]
   For both explicit and implicit treatment of Coriolis force, \( FX = 2.0 \times OMEGA \times SINEA_{i,j}. \)

5. Add in pressure term, masking out land with the GM masking array.
   \[ UA_{i,k} = GM_{i,k} \times (UA_{i,k} - DPDX_{i,k}) \]
   \[ VA_{i,k} = GM_{i,k} \times (VA_{i,k} - DPDY_{i,k}). \]

6. Form time change of vertically averaged forcing.
   a. Integrate time change vertically:
      \[ ZUN_i = ZUN_i + UA_{i,k} \times FX \]
      \[ ZVN_i = ZVN_i + VA_{i,k} \times FX, \]
      where \( FX = C2DTSF \times DZ_k. \)
   b. Form average by multiplying by reciprocal of depth (dividing by depth):
      \[ ZUN_i = ZUN_i \times HR_{i,j} \]
      \[ ZVN_i = ZVN_i \times HR_{i,j}. \]

7. Do analysis of internal mode forcing on energy timestep. Form vertical average for use later in external mode analysis.
a. Compute change in kinetic energy due to pressure term:

\[
\begin{align*}
UENG_{i,k} &= GM_{i,k} * - DPDX_{i,k}, \\
VENG_{i,k} &= GM_{i,k} * - DPDY_{i,k},
\end{align*}
\]

\[
ENGINT_6 = ENGINT_6 + (USA\!N_{i,k} * UENG_{i,k} + VSA\!N_{i,k} * VENG_{i,k}) * FX * DXU_i * DZ_k,
\]

where \( FX = CS_j * DYU_j \),

\[
ZUNENG_{i,6} = ZUNENG_{i,6} + UENG_{i,k} * DZ_k * HR_{ij},
\]

\[
ZVNENG_{i,6} = ZVNENG_{i,6} + VENG_{i,k} * DZ_k * HR_{ij}.
\]

b. Compute change in kinetic energy due to advection of momentum.

\[
UENG_{i,k} = GM_{i,k} * ((-FUW_{i+1,k} * (U_{i+1,k} + U_{i,k}) + FUW_{i,k} * (U_{i,k} + U_{i-1,k})
* DXU2R_i - FVv_{i,k} * (UP_{i,k} + U_{i,k}) + FVSU_{i,k} * (U_{i,k} + UM_{i,k}))
\]

\[
VENG_{i,k} = GM_{i,k} * ((-FUW_{i+1,k} * (V_{i+1,k} + V_{i,k}) + FUW_{i,k} * (V_{i,k} + V_{i-1,k})
* DXU2R_i - FVv_{i,k} * (VP_{i,k} + V_{i,k}) + FVSU_{i,k} * (V_{i,k} + VM_{i,k}))
\]

\[
ENGINT_2 = ENGINT_2 + (USA\!N_{i,k} * UENG_{i,k} + VSA\!N_{i,k} * VENG_{i,k}) * FX * DXU_i * DZ_k
\]

\[
ZUNENG_{i,2} = ZUNENG_{i,2} + UENG_{i,k} * DZ_k * HR_{ij},
\]

\[
ZVNENG_{i,2} = ZVNENG_{i,2} + VENG_{i,k} * DZ_k * HR_{ij}.
\]

Second Loop:

\[
UENG_{i,k} = GM_{i,k} * (-W_{i,k} * (U_{i,k-1} + U_{i,k}) - W_{i,k+1} * (U_{i,k} + U_{i,k+1})) * DZ2RQ_{i,k}
\]

\[
VENG_{i,k} = GM_{i,k} * (-W_{i,k} * (V_{i,k-1} + V_{i,k}) - W_{i,k+1} * (V_{i,k} + V_{i,k+1})) * DZ2RQ_{i,k}
\]

\[
ENGINT_3 = ENGINT_3 + (USA\!N_{i,k} * UENG_{i,k} + VSA\!N_{i,k} * VENG_{i,k}) * FX * DXU_i * DZ_k
\]

\[
ZUNENG_{i,3} = ZUNENG_{i,3} + UENG_{i,k} * DZ_k * HR_{ij},
\]

\[
ZVNENG_{i,3} = ZVNENG_{i,3} + VENG_{i,k} * DZ_k * HR_{ij}.
\]

c. Compute change in kinetic energy due to horizontal diffusion of momentum:

\[
UENG_{i,k} = GM_{i,k} * (BBUJ * DXU2R_i * (DXTAR_{i+1} * (UB_{i+1,k} - UB_{i,k})
+ DXTAR_i * (UB_{i-1,k} - UB_{i,k})) + CCUJ * (UB_{i,k} - UB_{i,k})
+ DDUJ * (UBM_{i,k} - UB_{i,k}) + GGUJ * UB_{i,k} - HHUJ
* DXU2R_i * (VB_{i+1,k} - VB_{i-1,k}))
\]

\[
VENG_{i,k} = GM_{i,k} * (BBUJ * DXU2R_i * (DXTAR_{i+1} * (VB_{i+1,k} - VB_{i,k})
+ DXTAR_i * (VB_{i-1,k} - VB_{i,k})) + CCUJ * (VB_{i,k} - VB_{i,k})
+ DDUJ * (VBM_{i,k} - VB_{i,k}) + GGUJ * VB_{i,k} + HHUJ
* DXU2R_i * (UB_{i+1,k} - UB_{i-1,k}))
\]

\[
ENGINT_4 = ENGINT_4 + (USA\!N_{i,k} * UENG_{i,k} + VSA\!N_{i,k} * VENG_{i,k}) * FX * DXU_i * DZ_k
\]

\[
ZUNENG_{i,4} = ZUNENG_{i,4} + UENG_{i,k} * DZ_k * HR_{ij},
\]

\[
ZVNENG_{i,4} = ZVNENG_{i,4} + VENG_{i,k} * DZ_k * HR_{ij}.
\]

d. Compute change in kinetic energy due to wind stress:

\[
UENG_{i,1} = GM_{i,1} * EEM_1 * (UOVER_i - UdIF_{i,1})
\]

\[
VENG_{i,1} = GM_{i,1} * EEM_1 * (VOVER_i - VDIF_{i,1})
\]
\[ \text{ENGINT}_7 = \text{ENGIT}_7 + (\text{USAV}_{i,1} \times \text{UENG}_{i,1} + \text{VSAV}_{i,1} \times \text{VENG}_{i,1}) \times \text{FX} \times \text{DXU}_{i} \times \text{DZ}_1 \]
\[ \text{ZUNENG}_{i,7} = \text{ZUNENG}_{i,7} + \text{UENG}_{i,1} \times \text{DZ}_1 \times \text{HR}_{ij} \]
\[ \text{ZVNENG}_{i,7} = \text{ZVNENG}_{i,7} + \text{VENG}_{i,1} \times \text{DZ}_1 \times \text{HR}_{ij} \]

e. Compute change in kinetic energy due to bottom drag:
\[ \text{UENG}_{ikz} = \text{GM}_{ikz} \times \text{FFM}_{kz} \times (\text{UDIF}_{ikz+1} - \text{UDIF}_{ikz}) \]
\[ \text{VENG}_{ikz} = \text{GM}_{ikz} \times \text{FFM}_{kz} \times (\text{VDIF}_{ikz+1} - \text{VDIF}_{ikz}) \]
\[ \text{ENGINT}_8 = \text{ENGINT}_8 + (\text{USAV}_{ikz} \times \text{UENG}_{ikz} + \text{VSAV}_{ikz} \times \text{VENG}_{ikz}) \times \text{FX} \times \text{DXU}_{i} \times \text{DZ}_k \]
\[ \text{ZUNENG}_{i,8} = \text{ZUNENG}_{i,8} + \text{UENG}_{ikz} \times \text{DZ}_k \times \text{HR}_{ij} \]
\[ \text{ZVNENG}_{i,8} = \text{ZVNENG}_{i,8} + \text{VENG}_{ikz} \times \text{DZ}_k \times \text{HR}_{ij} \]

where \( KZ \) is equal to the index of the bottom level.

f. Compute change in kinetic energy due to vertical diffusion of momentum:
\[ \text{UENG}_{i,k} = \text{GM}_{i,k} \times (\text{FXA} \times \text{EEM}_k \times (\text{UDIF}_{ik}-1 - \text{UDIF}_{ik}) - \text{FXB} \times \text{FFM}_k \times (\text{UDIF}_{ik} - \text{UDIF}_{i,k+1})) \]
\[ \text{VENG}_{i,k} = \text{GM}_{i,k} \times (\text{FXA} \times \text{EEM}_k \times (\text{VDIF}_{ik}-1 - \text{VDIF}_{ik}) - \text{FXB} \times \text{FFM}_k \times (\text{VDIF}_{ik} - \text{VDIF}_{i,k+1})) \]
\[ \text{ENGINT}_5 = \text{ENGINT}_5 + (\text{USAV}_{ik} \times \text{UENG}_{ik} + \text{VSAV}_{ik} \times \text{VENG}_{ik}) \times \text{FX} \times \text{DXU}_{i} \times \text{DZ}_k \]
\[ \text{ZUNENG}_{i,5} = \text{ZUNENG}_{i,5} + \text{UENG}_{ik} \times \text{DZ}_k \times \text{HR}_{ij} \]
\[ \text{ZVNENG}_{i,5} = \text{ZVNENG}_{i,5} + \text{VENG}_{ik} \times \text{DZ}_k \times \text{HR}_{ij} \]


For explicit treatment:
\[ \text{UA}_{i,k} = \text{UB}_{i,k} + \text{C2DTUV} \times \text{UA}_{i,k} \]
\[ \text{VA}_{i,k} = \text{VB}_{i,k} + \text{C2DTUV} \times \text{VA}_{i,k} \]

For implicit treatment:
\[ \text{UDIF}_{i,k} = (\text{UA}_{i,k} + \text{FX} \times \text{VA}_{i,k}) \times \text{DETMR} \]
\[ \text{VDIF}_{i,k} = (\text{VA}_{i,k} - \text{FX} \times \text{UA}_{i,k}) \times \text{DETMR} \]
where \( \text{FX} = \text{C2DTUV} \times \text{ACOR} \times 2.0 \times \text{OMEGA} \times \text{SINEA}_{ij} \) and \( \text{DETMR} = 1.0/(1.0 + \text{FX}^2) \).
\[ \text{UA}_{i,k} = \text{UB}_{i,k} + \text{C2DTUV} \times \text{UDIF}_{i,k} \]
\[ \text{VA}_{i,k} = \text{VB}_{i,k} + \text{C2DTUV} \times \text{VDIF}_{i,k} \]

9. Determine the incorrect vertical means of the new velocities.

First, sum the new velocities multiplied by the vertical grid size of each cell in centimeter
\[ \text{SFU}_i = \text{SFU}_i + \text{UA}_{i,k} \times \text{DZ}_k \]
\[ \text{SFV}_i = \text{SFV}_i + \text{VA}_{i,k} \times \text{DZ}_k. \]
Find the means by dividing each column sum by depth (multiplying by the reciprocal):

\[ SFU_i = SFU_i * HR_{ij} \]

\[ SFV_i = SFV_i * HR_{ij} \]

10. Subtract incorrect vertical means from newly computed velocities to get correct internal mode velocity.

11. If this is an energy timestep, compute total change of kinetic energy of internal mode:

\[ ENGINT_1 = ENGINT_1 + (USA_{i,k} * (UA_{i,k} - UB_{i,k}) + VSA_{i,k} * (VA_{i,k} - VB_{i,k})) * FX * DXU_i \]

where \( FX = CS_{j} * DYU_{j} * DZ_{k} / (2DTUV) \).

**Begin computation of vorticity for input to CSU CRELAX:**

1. Form curl of time change in vertically averaged equations:

\[ ZTD_{ij} = ((ZUN_i * DXU_i + ZUN_{i-1} * DXU_{i-1}) * CS_j - (ZUS_i * DXU_i + ZUS_{i-1} * DXU_{i-1}) * CS_{j-1}) \]

\[ ZTD_{ij} = (((ZVN_i - ZVN_{i-1}) * DYU_j + (ZVS_i - ZVS_{i-1}) * DYU_{j-1} - ZTD_{ij}) * DXT2R_{i} * DYT_{j}) * CSTR_j \]

2. If on an energy timestep, do analysis of external mode forcing:

\[ ENGEXT_{i,ll} = ENGEXT_{i,ll} - P_{ij} * (((ZVEN_{i,ll} - ZVEN_{i-1,ll}) * DYU_j + (ZVS_{i,ll} - ZVS_{i-1,ll}) * DXU_{j-1} * CS_{j} - (ZUSEN_{i,ll} * DXU_{i} + ZUSEN_{i-1,ll} * DXU_{i-1}) * CS_{j-1}) * DYT_{2R_{i}} * DXT_{j}) \]

Transfer computed quantities to the north of the present row to be defined to the south in the computation for the next row.

### 3.4.4 CSC Compute Tracers

CSC Compute Tracers must compute the tracer elements of temperature and salinity and be readily adaptable to computing additional tracers.

#### 3.4.4.1 CSU TRACER

CSU TRACER meets the requirements of CSC Compute Tracers by computing temperature and salinity and is designed to be adaptable to computing additional tracer elements.

#### 3.4.4.1.1 CSU TRACER Design Specification/Constraints – There are no known constraints.

#### 3.4.4.1.2 CSU TRACER Design

**Input Data Elements:**

/COX2/

FW1    heat above the freezing temperature
GICE   ice growth rate computed by the ice model
SHICE growth rate of ice thickness

/FULLWD/
BUOY energy transfer through buoyancy effects
DTABS volume average of absolute change of temperature
ITT timestep counter; total number of timesteps completed
KMT number of vertical levels of the ocean at T points
MXP if = 1, indicates second pass of a Euler backward timestep
ENERGY if = 1, indicates energy printout timestep
NTSI number of timesteps between prints of a single line of data
TTDTOT array of integrals on tracers
TVAR change of variance of tracers

/ONEDIM/
CS cosine of U,V point latitudes
CSTR reciprocal of cosine of T point latitudes
DXT zonal grid spacing across T boxes (between U,V points)
DXT4R reciprocal of DXT × 4.0
DXU2R reciprocal of DXU × 2.0
DYT meridional grid spacing across T boxes (between U,V points)
DYTR reciprocal of meridional grid spacing across T boxes
DYU meridional grid spacing across U,V boxes (between T points)
DYUR reciprocal of meridional grid spacing across U,V boxes
DZ vertical layer thickness in centimeters
DZZ vertical grid spacing between U,V,T points
DZZ2R 1.0/(DZZ * 2.0)

/RSTRRT/
AREA1 fraction of the grid cell covered with ice
HEFF mean ice thickness per grid cell

/RVR/
IRIVER x component of river discharge
JRIVER y component of river discharge
KRIIVER z component of river discharge
RIVER river discharge temperature

/SCALAR/
AH coefficient of horizontal mixing of T
C2DTTS two times the length of the timestep on T
DTTS length of the timestep on T
GRAV acceleration due to gravity; = 980.6 cm/s²

/TSTOP/
CHENG

/WORKSP1/
C2DZQ vertical layer thickness in centimeters × 2; for future vectorization
DXT4RQ 1.0/(DXT × 4.0); for future vectorization
DXTQ zonal grid spacing across T boxes; for future vectorization
DXU2RQ 1.0/(DXU × 2.0); for future vectorization
DXUQ: zonal grid spacing across U,V boxes; for future vectorization
DZ2RQ: 1.0/(DZ * 2.0); for future vectorization
EEHQ: upper vertical mixing coefficient of T; for future vectorization
FFHQ: lower vertical mixing coefficient of T; for future vectorization
FM: masking array for T points; 0 indicates land points, 1 indicates ocean points
FMM: FM at row J - 1
FMP: FM at row J + 1
FVST: advective coefficient for south face of T box
RHON: density with a space and time invariant subtracted for the row to the north
       of the present row
RHOS: density with a space and time invariant subtracted for the row to the south
       of the present row
T: tracer variables for current timestep, row J
TB: tracer for timestep before present
TBM: similar to TB but in row J - 1
TBP: similar to TB but in row J + 1
TM: tracer in row J - 1 for present timestep
TP: tracer in row J + 1 for present timestep
UM: zonal component of velocity in row J - 1
V: meridional component of velocity

Output Data Elements:

/CXO2/
FW1: fraction of the grid cell covered by ice
GICE: ice growth rates computed by the ice portion of the PIPS2.0 coupled model

/FULLWD/
BUOY: energy transfer through buoyancy effects
DTABS: volume average of absolute change of temperature
TTDTOT: array of integrals on tracers
TVAR: change of variance of tracers

/WORKSPI/
FVST: advective coefficient for south face of T box
RHOS: density with a space and time invariant subtracted for the row to the south
       of the present row
T: tracer variables (temperature and salinity) for current timestep, row J
TA: tracer variables for timestep after present
TDIF: diffusion computation array
TEMPA: temporary storage
TEMPB: temporary storage
UA: zonal component of velocity for timestep after present
VA: meridional component of velocity for timestep after present

Parameters:
stored in file ocean.par
The ocean.par parameters are described in Sec. 3.4.1.1.2.
Logic Flow:

Prepare arrays for the computation of the tracers:

Several arrays must be calculated in order to prepare for the computation of the tracers:

- the advective coefficients for the west (\(FUW\)) and north (\(FVN\)) faces of the T box,
- vertical velocity in T columns, and
- boundary conditions for the vertical diffusion of tracers.

1. Find advective coefficients for west and north faces of T box:
\[
FUW_{i,k} = (U_{i-1,k} \ast DYU_j + U_{i-1,k} \ast DYU_{j-1}) \ast FXA
\]
\[
FVN_{i,k} = (V_{i,k} \ast DXUQ_{i,k} + V_{i-1,k} \ast DXUQ_{i-1,k}) \ast FXB \ast DXT4RQ_{i,k}.
\]

2. Compute vertical velocity in T columns:

a. Set vertical velocity at the surface to 0 (rigid-lid assumption).

b. Compute change of vertical velocity between levels:
\[
W_{i,k+1} = C2DZQ_{i,k} \ast ((FUW_{i+1,k} - FUW_{i,k}) \ast DXT4RQ_{i,k} + FVN_{i,k} - FVST_{i,k}).
\]
c. Integrate downward from the surface by adding successive vertical velocities.

3. Set boundary conditions for vertical diffusion of tracers:

a. Transfer interior points (\(TB\)) into diffusion computation array (\(TDIF\)).

b. Set top point of the column to reflect surface flux, bottom point of the column to reflect insulation:
\[
TDIF_{i,1,m} = TB_{i,1,m}
\]
\[
TDIF_{i,kz+2,m} = TB_{i,kz+2,m}.
\]

Compute the tracers:

Implement the equations given in Sec. 2.2.4.3 for computing tracer elements. Two passes are required: in the first pass, temperature is computed; in the second pass, salinity is computed.

1. Compute total advection of tracers by summing the flux through the west face of the T box, the zonal flux divergence, the meridional flux divergence, the flux through the top of the T box, and the vertical flux divergence.

a. Compute flux through west face of T box:
\[
TEMPA_{i,k} = FUW_{i,k} \ast (T_{i,km} + T_{i-1,k,m})
\]
\[
TA_{i,1,1} = MAX(TA_{i,1,1} - 54.4 \ast (TA_{i,1,2} + 0.035))
\]
\[
TMPCHG_{i,k} = FUW_{i,k} \ast T_{i,km} + T_{i-1,k,m} + 2.0 \ast TFRG.
\]

b. Compute zonal flux divergence for all grid cells in slab j:
\[
TA_{i,km} = (TEMPA_{i,k} - TEMPA_{i+1,k}) \ast DXT4RQ_{i,k}.
\]
For surface temperature,
\[
TCHENG_{ij} = (TMPCHG_{i,k} - TMPCHG_{i+1,k}) \ast DXT4RQ_{i,k}.
\]
c. Add in meridional flux divergence:

\[ TA_{ik,m} = TA_{ik,m} - FVN_{ik} \ast (TP_{ik,m} + T_{ik,m}) + FVST_{ik} \ast (T_{ik,m} + TM_{ik,m}) \]

For surface temperature,

\[ TCHENG_{ij} = TCHENG_{ij} - FVN_{ik} \ast (TP_{ik,m} + T_{ik,m}) + FVST_{ik} \ast (T_{ik,m} + TM_{ik,m}). \]

d. Compute flux through top of T box:

\[ TEMPB_{ik} = W_{ik} \ast (T_{ik-1,m} + T_{ik,m}) \]

For surface temperature,

\[ TMPCHG_{ik} = W_{ik} \ast (T_{ik-1,m} + T_{ik,m} + 2.0 \ast TFRG). \]

e. Add in vertical flux divergence:

\[ TA_{ik,m} = TA_{ik,m} + (TEMPB_{ik+1} - TEMPB_{ik}) \ast DZ2RQ_{ik} \]

For surface temperature,

\[ TCHENG_{ij} = TCHENG_{ij} + (TMPCHG_{ik+1} - TMPCHG_{ik}) \ast DZ2RQ_{ik}. \]

2. Add horizontal diffusion of tracers (evaluated at tau \(- 1\) timestep) to the total advection of the tracers.

a. Compute coefficients dependent only on latitude:

\[ BBTJ = 8.0 \ast AH \ast CSTR_j^2 \]
\[ CCTJ = AH \ast CS_j \ast DYUR_j \ast DYTR_j \ast CSTR_j \]
\[ DDTJ = AH \ast CS_{j-1} \ast DYUR_{j-1} \ast DYTR_j \ast CSTR_j \]

b. Compute gradients at west face of T box:

\[ TEMPA_{ik} = DXU2RQ_{i-1,k} \ast (TB_{ik,m} - TB_{i-1,k,m}). \]

c. Add in final contribution from horizontal diffusion of tracers. To provide for insulated walls, each gradient is multiplied by the mask of the point in its respective direction causing it to be 0 if it is taken across a wall:

\[ TA_{ik,m} = TA_{ik,m} + BBTJ \ast DXT4RQ_{ik} \ast (FM_{i+1,k} \ast TEMPA_{i+1,k} - FM_{i-1,k} \ast TEMPA_{ik}) + CCTJ \ast FMP_{ik} \ast (TBP_{ik,m} - TB_{ik,m}) + DDTJ \ast FMM_{ik} \ast (TBM_{ik,m} - TB_{ik,m}). \]

3. Add vertical diffusion of tracers to the sum of the total advection of the tracers plus the horizontal diffusion of the tracers.

a. Compute gradients at top of T box:

\[ TEMPB_{ik} = TDIF_{ik,m} - TDIF_{ik+1,m}. \]

b. Add in final contribution from vertical diffusion of tracers:

\[ TA_{ik,m} = TA_{ik,m} + EEHQ_{ik} \ast TEMPB_{ik} - FFHQ_{ik} \ast TEMPB_{ik+1}. \]
4. Set Newtonian boundary condition of temperature and salinity at the ocean surface.

For surface grid cells:

\[ TA_{i,1,m} = TA_{i,1,m} + 4.63E-8 \times (CHENG_{i,j,m,1} - TB_{i,1,m}). \]

For surface temperature \((m=1):\)

\[ TA_{i,1,1} = TA_{i,1,1} - GICE_{i,j} \times RICE0/30.00, \]

where \(RICE0 = 302.0E6/4.19E6.\)

For surface salinity, in open water, the atmospheric heating should not increase or decrease the salinity because no ice melting is involved. Open-water grid cells are indicated by \(AICE(i,j,2) \leq 0.15, HICE(i,j,2) \leq 0.1,\) and \(SHICE(i,j) \leq 0.\) For open-water grid cells (surface level only) reset \(ASH\) to 0.

For surface salinity:

\[ TA_{i,1,2} = TA_{i,1,2} + SHICE \times 0.035 \times ASH/30.00. \]

For all other levels \((k \neq 1):\)

\[ TA_{i,k,m} = TA_{i,k,m} + 4.63E-8 \times (CHENG_{i,j,m,k} - TB_{i,k,m}). \]

5. Compute new tracers, resetting land points to 0 by multiplying by the land mask array, \(FM.\) New tracers are computed by multiplying the time rate of change of tracer values (computed in steps 1–4 above and represented by \(TA\)) by twice the timestep on \(T\) and adding this value to the tracer quantities computed for the timestep before present. Factor the \(FM\) array into the equation to force 0 values for array elements that represent land points.

6. Set salinity to 45 ppt over land to stop convection there. \(TA_{i,k,2} = 0.01\) is equal to 45 ppt, since model units are \((ppt - 35.0)/1000.0.\)

7. On an energy timestep, perform analysis of tracer forcing.

a. Compute change of tracer due to advection:

\[ TTDTOT_{2,m} = TTDTOT_{2,m} + BOXVOL \times (\text{\textminus}FUW_{i+1,k} \times (T_{i+1,k,m} + T_{i,k,m})
+ FUW_{i,k} \times (T_{i,k,m} + T_{i-1,k,m})) \times DTX4R_{i} - FVN_{i,k} \times (TP_{i,k,m} + T_{i,k,m})
+ FVST_{i,k} \times (T_{i,k,m} + TM_{i,k,m})). \]

b. Compute change of tracer due to horizontal diffusion:

\[ TTDTOT_{4,m} = TTDTOT_{4,m} + BOXVOL \times (BBTJ \times DXU2R_{i} \times DTX4R_{i} \times FM_{i+1,k}
+ (TB_{i+1,k,m} - TB_{i,k,m}) + BBTJ \times DXU2R_{i-1} \times DTX4R_{i} \times FM_{i-1,k}
+ (TB_{i-1,k,m} - TB_{i,k,m}) + CCTJ \times FMP_{i,k} \times (TBP_{i,k,m} - TB_{i,k,m})
+ DDTJ \times FMM_{i,k} \times (TBM_{i,k,m} - TB_{i,k,m})). \]

c. Compute change of tracer due to vertical diffusion:

\[ TTDTOT_{5,m} = TTDTOT_{5,m} + BOXVOL \times (EEH_{k} \times (TDIF_{i,k,m} - TDIF_{i,k+1,m}) - FFH_{k}
\times (TDIF_{i,k+1,m} - TDIF_{i,k+2,m})). \]

d. Add in contribution from Newtonian boundary condition of temperature and salinity at the ocean surface.
For surface temperature:

\[ TTDTOT_{5,1} = TTDTOT_{5,1} + 4.63 \times 10^{-8} \times (CHENG_{i,j,m,k} - TB_{i,k,m}) \times DZ_k \]
\[ \text{GICE}_{i,j} \times \text{RICE0} \times DZ_k/30.00. \]

For surface salinity:

\[ TTDTOT_{5,2} = TTDTOT_{5,2} + 4.63 \times 10^{-8} \times (CHENG_{i,j,m,k} - TB_{i,k,m}) \times DZ_k \]
\[ + \text{SHICE}_{i,j} \times 0.035 \times \text{ASH} \times DZ_k/30.00, \]

for other than surface grid cells (k ≠ 1).

For temperature:

\[ TTDTOT_{5,1} = TTDTOT_{5,1} + 4.63 \times 10^{-8} \times (CHENG_{i,j,m,k} - TB_{i,k,m}) \times DZ_k. \]

For salinity:

\[ TTDTOT_{5,2} = TTDTOT_{5,2} + 4.63 \times 10^{-8} \times (CHENG_{i,j,m,k} - TB_{i,k,m}) \times DZ_k. \]

e. Compute total energy exchange between potential and kinetic:

\[ Buoy = Buoy - FX \times DZZ_k \times W_{i,k} \times (RHOS_{i,k-1} + RHOS_{i,k}) \]

8. Convectively adjust water column if gravitationally unstable. Allow for variable number of passes through convection loop using an outer loop with limits from 1 to NCON where NCON is presently hardwired to 1.

a. Call CSU STATEC to compute density for entire slab to determine stability.

b. For each tracer, mix adjoining levels if unstable. Tracers are unstable if density of a grid cell is greater than the density of the grid cell below it. To mix unstable grid cells:

\[ TA_{i,k,m} = (DZ_k \times TA_{i,k,m} + DZ_{k+1} \times TA_{i,k+1,m}) \times DZZ2R_{k+1}. \]

Integrate changes and prepare for next timestep:

1. Integrate total changes in T, S and squared T, S on energy timestep.

2. Accumulate integrated absolute changes in T every NTSI timesteps.

3. Transfer quantities computed to the north of the present row to be defined to the south in the computation of the next row.

4. Set new velocities at northern wall to 0 since no pass through CSU CLINIC is made for this row.

3.4.5 CSC Compute External Mode

The external mode of velocity must be computed in terms of a mass transport stream function.

3.4.5.1 CSU CRELAX

CSU CRELAX is called once at the end of each timestep by CSU OSTEP. It takes the vorticity function computed in CSU CLINIC and, using sequential overrelaxation, solves the LaPlacian equation for the external mode of velocity in terms of a mass transport stream function.
3.4.5.1.1 CSU CRELAX Design Specification/Constraints – There are no known constraints.

4.4.5.1.2 CSU CRELAX Design

**Input Data Elements:**

/FIELDS/

HR  reciprocal of total depth at U,V point  
P   mass transport stream function  
PB  mass transport stream function for previous timestep  
ZTD change of vorticity across one timestep  

/FULLWD/

EB  if true: the current step is a Euler backward timestep  
IEIS array of I ending indices for island boxes  
IEZ  array of ending indices for vorticity  
ISIS array of I starting indices for island boxes  
ISZ  array of starting indices for vorticity  
ITT  timestep counter  
JEIS array of J ending indices for island boxes  
JSIS array of J starting indices for island boxes  
KFLDS disk unit number for two-dimensional horizontal fields and start and end indices; = 12  
MIX  mixing timestep indicator; set equal to 1 on a mixing timestep  
MXP  if = 1, second pass of a Euler backward timestep  
MXSCAN maximum number of scans allowed for convergence  

/ONEDIM/

CS  cosine of U,V point latitudes  
CST  cosine of T point latitudes  
CSTR reciprocal of cosine of T point latitudes  
DXT  zonal grid spacing across T boxes  
DXTR 1.0/DXT  
DXUR 1.0/DXU  
DYT  meridional grid spacing across T boxes  
DYTR reciprocal of meridional grid spacing across T boxes (between U,V points)  
DYUR reciprocal of meridional grid spacing across U,V boxes (between T points)  
SINEA sine of U,V point latitude in the Earth-oriented coordinates; used to compute Coriolis force  

/SCALAR/

ACOR if > 0, treat the Coriolis term implicitly with forward component weighted by ACOR, past component by 1 – ACOR  
C2DTSF DTSF × 2  
CRIT criterion for convergence of relaxation  
OMEGA rate of rotation of the coordinate system  
SOR coefficient of overrelaxation  

**Output Data Elements:**

/FIELDS/

P  mass transport stream function  
PB  mass transport stream function for previous timestep
/FULLWD/
   MSCAN   scan counter

/WORKSP2/
   CFE       coefficient of eastern point in LaPlacian star
   CFN       coefficient of northern point in LaPlacian star
   CFS       coefficient of southern point in LaPlacian star
   CFW       coefficient of western point in LaPlacian star
   COF       normalization array in computation of island flow
   COFIS     integral of COF
   CPF       normalization factor used in constructing LaPlacian star
   ISMASK    gridpoint type indicator: = 0 over interior points; = 1 over perimeter points;
             = 2 over land points
   PTD       change of stream function across a timestep
   PTDB      change of stream function across previous timestep
   RES       residual of relaxation

Parameters:
stored in file ocean.par
   The ocean.par parameters are described in Sec. 3.4.1.1.2.

Logic Flow:
Prepare for the relaxation:
   Begin introductory section to prepare for the relaxation:

1. Initialize working arrays.

2. Read in the relaxation solution from the timestep before present and the relaxation solution of
   the present timestep.

3. Form island mask by distinguishing interior ocean points, perimeter ocean points, and land
   points.

4. Calculate the depth field from the depth field reciprocal array.

5. Ensure that all points over land are exactly 0.

6. Generate arrays of coefficients for relaxation.

   a. Compute coefficients of the LaPlacian star, augment coefficients for implicit treatment of
      Coriolis term, and normalize.

      Northern point correction coefficient:
      \[ CFN_{i,j} = 2.0 \times CS_j \times CSTR_j \times DYTR_j \times DYUR_j / (H_{i-1,j} + H_{i,j}) \]
      \[ CFN_{i,j} = CFN_{i,j} + (HR_{i,j} - HR_{i-1,j}) \times SINEA_{i,j} \times FX \times DXTR_i \]
      \[ CFN_{i,j} = CFN_{i,j} \times CPF_i. \]

      Southern point correction coefficient:
      \[ CFS_{i,j} = 2.0 \times CS_{j-1} \times CSTR_j \times DYTR_j \times DYUR_{j-1} / (H_{i-1,j-1} + H_{i,j-1}) \]
\[ CFS_{ij} = CFS_{ij} - (HR_{ij-1} - HR_{i-1,j-1}) \times SINEA_{ij-1} \times FX \times DXTR_i \]
\[ CFS_{ij} = CFS_{ij} \times CPF_i. \]

**Eastern point correction coefficient:**
\[ CFE_{ij} = 2.0 \times CSTR_j^2 \times DXUR_i \times DXTR_i/(H_{ij} + H_{ij-1}) \]
\[ CFE_{ij} = CFE_{ij} - (HR_{ij} \times SINEA_{ij} - HR_{ij-1} \times SINEA_{ij-1}) \times FX \times DXTR_i \]
\[ CFE_{ij} = CFE_{ij} \times CPF_i. \]

**Western point correction coefficient:**
\[ CFW_{ij} = 2.0 \times CSTR_j^2 \times DXUR_{i-1} \times DXTR_i/(H_{i-1,j} + H_{i-1,j-1}) \]
\[ CFW_{ij} = CFW_{ij} + (HR_{i-1,j} \times SINEA_{ij} - HR_{i-1,j-1} \times SINEA_{ij-1}) \times FX \times DXTR_i \]
\[ CFW_{ij} = CFW_{ij} \times CPF_i, \]
where \( FX = -C2DSF \times ACOR \times CSTR_j \times DYT R_j \times OMEGA. \)

b. Compute coefficients on island perimeter points. Identify islands according to start and stop island indices stored in ISIS, IEIS, JSIS, and JEIS for ISMASK array. A value of 1 is used to indicate island perimeter points in the ISMASK array.

Island coefficients are computed based on which side of the grid cell is adjacent to land (identified by \( HR = 0 \)).

Assign northern coefficient if \( HR_{i-1,j} \) is not 0 or \( HR_{ij} \) is not 0:
\[ CFN_{ij} = (2.0 \times CS_j \times DYUR_j \times DYT R_j \times CSTR_j)/(PTD_{i-1,j} + PTD_{ij}) \]
\[ + FX \times DXTR_i \times (HR_{ij} - HR_{i-1,j}) \times SINEA_{ij}. \]

Assign southern coefficient if \( HR_{i-1,j-1} \) is not 0 or \( HR_{ij-1} \) is not 0:
\[ CFN_{ij} = FX/(PTD_{i-1,j-1} + PTD_{ij-1}) - FXD \times DXTR_i \times (HR_{ij-1} - HR_{i-1,j-1}) \times SINEA_{ij-1}. \]

Assign eastern coefficient if \( HR_{ij} \) is not 0 or \( HR_{ij-1} \) is not 0:
\[ CFE_{ij} = FXA \times DXTR_i \times DXUR_j/(PTD_{ij} + PTD_{ij-1}) - FXD \times DXTR_i \]
\[ \times (HR_{ij} \times SINEA_{ij} - HR_{ij-1} \times SINEA_{ij-1}). \]

Assign western coefficient if \( HR_{i-1,j} \) is not 0 or \( HR_{i-1,j-1} \) is not 0:
\[ CFW_{ij} = FXA \times DXTR_i \times DXUR_{i-1}/(PTD_{i-1,j} + PTD_{i-1,j-1}) + FXD \times DXTR_i \]
\[ \times (HR_{i-1,j} \times SINEA_{ij} - HR_{i-1,j-1} \times SINEA_{ij-1}). \]

Multiply all island perimeter coefficients \( CFN, CFS, CFE, \) and \( CFW \) by \( COF \) normalization array.

Sum \( COFIS \) over all island grid boxes.

c. Multiply all of the \( CFN, CFS, CFE, \) and \( CFW \) coefficients by the overrelaxation factor \( SOR \) read in from NAMELIST PARMS. Also set \( COFIS \) for the island equal to its reciprocal.
7. Compute a first guess for the relaxation by extrapolating the two previous solutions forward in time:
   a. Finish reading in the relaxation solution of the previous timestep, array PTD.
   b. Perform time extrapolation, accounting for the mixing timestep.
      \[ PTD_{i+1,j} = FXA \times (2.0 \times PTD_{i,j} - PTD_{i-1,j}), \]
      where \( FXA = 0.5 \) on a mixing timestep and 1.0 otherwise.
8. Compute criterion for the convergence of relaxation: \( CRTP = CRIT \times FXA \times SOR. \)
9. Initialize residuals array \( RES \) to 0.

**Loop to compute the relaxation:**

This algorithm uses the method of successive overrelaxation where a guess is made for all grid cells variant in \( P \) (the stream function), a residual is computed based on the stream function and the boundary conditions, and a new guess is established from the residual. This process is iterated until the change in \( P \) between guesses is less than the specified criterion.

1. Compute entire field of residuals as in simultaneous relaxation:
   \[ RES_{i,j} = CFN_{i,j} \times PTD_{i+1,j} + CFS_{i,j} \times PTD_{i,j-1} + CFE_{i,j} \times PTD_{i+1,j} + CFW_{i,j} \]
   \[ \times PTD_{i-1,j} - SOR \times (PTD_{i,j} + ZTD_{i,j}). \]
2. Reset residuals over land to 0 based on start and end indices in arrays ISZ and IEZ.
3. Perform correction on southern point to yield sequential relaxation using correction factor computed in the initialization procedure:
   \[ RES_{i,j} = RES_{i,j} + CFS_{i,j} \times RES_{i,j-1}. \]
4. Perform correction on western point to yield sequential relaxation using correction factor computed in the initialization procedure:
   \[ RES_{i,j} = RES_{i,j} + CFW_{i,j} \times RES_{i,j-1}. \]
5. Correct change of stream function based on residuals by adding \( RES_{i,j} \) to \( PTD_{i,j} \).
6. Find the maximum absolute residual to determine convergence.
7. Do hole relaxation for each island and loop for the number of islands in the model basin.
   a. Compute island residuals:
      Initialize island residual (\( RESIS \)) to 0.
      Loop for the meridional start to stop island grid boxes (index \( J \)).
      Loop for the zonal start to stop island grid boxes (index \( I \)).
      If \( ISMASK_{i,j} \) equals 1, increment RESIS for this grid box:
      \[ RESIS = RESIS + (CFN_{i,j} \times PTD_{i,j+1} + CFS_{i,j} \times PTD_{i,j-1} + CFE_{i,j} \times PTD_{i+1,j} + CFW_{i,j} \]
      \[ \times PTD_{i-1,j} - SOR \times (PTD_{i,j} + ZTD_{i,j}) \] \( \times COF_{i,j}. \)
b. Normalize the island residual by its coefficient COFIS and update the maximum absolute residual of the relaxation if necessary.

c. Correct change of stream function over the island and its perimeter points by adding the island residual to it.

8. Test the maximum residual for convergence of the relaxation. If not converged, proceed with another scan by returning to step 1. If the maximum number of scans has been reached, accept the solution and proceed to update the stream function.

Update the stream function based on the relaxation solution:

1. Update the stream function based on the relaxation solution. On the second pass of the Euler backward timestep, update only the stream function for the present timestep \( P \), since the stream function for the timestep before present \( PB \) was updated on the first pass. If the current timestep is not the second pass of a Euler backward timestep, update the stream functions for the present timestep and the timestep before present.

2. Save change of stream function to compute first guess for relaxation next timestep. Bypass this section on the first pass of an Euler backward timestep, since it will be done on the second pass. Multiply the change of stream function by 2 if on a mixing timestep or if on the second pass of an Euler backward timestep.

3.4.6 CSC Direct Access Manager

The ocean model requires a set of routines that interface it with the local system I/O facility.

3.4.6.1 CSU ODAM

CSU ODAM consists of several entry points that collectively fulfill the requirements of CSC Direct Access Manager.

3.4.6.1.1 CSU ODAM Design Specification/Constraints – CSU ODAM will consist of several entry points instead of subroutines because of the close relationships between the I/O routines. For the core-contained mode, the routines contained in CSU ODAM must manage the transfer of data as if they resided in different files. In actuality, data for five logical units will all be stored in the same array (virtual disk). Logical units LABS(1), LABS(2), and LABS(3) (13,14, and 15, respectively) will be used for primary slab data. Logical unit KFLDS (set to 12) will be used for two-dimensional horizontal fields and logical unit KONTRL (set to 11) will be used for the timestep counter, total elapsed time, and the area and volume of the basin. Data stored in each of these units will be stored in an array in contiguous slots and in ascending order on logical unit number (e.g., data for unit 1 will be located in the first slot, data for unit 12 in the second, and so forth).

3.4.6.1.2 CSU ODAM Design

Input Data Elements:

- A: origination/destination array in memory
- LO: I/O unit number
- LU: I/O unit number
- NBLK: length of each block on the unit
- NBUF: number of buffers supplied to the unit
- NFRST: unit address of the first word to be transferred
- NTOT: length of unit in words
NWRS  number of words to transfer
BIG   read from tape to initialize virtual disk and later stored to tape to save contents of virtual disk

Output Data Elements:
A     origination/destination array in memory
BIG   read from tape to initialize virtual disk and later stored to tape to save contents of virtual disk

Contents of array BIG and more detailed ODAM I/O virtual disk access information is contained in Sec. 3.4.6.

Local Data Elements:
N11  20
N12  907200
NE   address of the last word to retrieve from array C
NS   address of the first word to retrieve from array C
NSKP number of words to skip on retrieval from array C
NSL  8035200
NTB  16977620

Parameters:
stored in file ocean.par
The ocean.par parameters are described in Sec. 3.4.1.1.2.

stored in file odam.par
N11  20
N12  907200
NSL  8035200
NTB  16977620

Logic Flow:
Define several entry points to simulate direct access input/output:

ENTRY OGET
ENTRY OGET will retrieve a variable number of words from array C and store them in array A. For \( LU > 11 \), \( N11 \) words are skipped (represented by \( NSKP \)). For \( LU > 12 \), \( N12 \) additional words are skipped. For the first word retrieved \( (NS) \), \( NS = NFRST + NSKP \) for logical units \( = 11 \) or \( 12 \). The first word retrieved for logical units \( \geq 13 \) is computed as follows:

\[
NS = NSKP + (2 \times \frac{(NFRST - 1) + NWRS}{LU - 13}) \times NWRS + 1.
\]

The last word retrieved \( (NE) = NS + NWRS - 1 \). In a core-contained mode, data is retrieved from array C for elements \( NS \) through \( NE \) and assigned to array A to be sent back to the calling routine.

ENTRY OPUT
ENTRY OPUT transfers a variable number of words from array A in memory and stores them in array C on virtual disk. The starting location for data storage is first determined by the logical unit number \( LU \) and the variable \( NFRST \) passed in the argument list. For logical unit numbers >11,
the number of words to skip (NSKP) is initialized to N11. For logical unit numbers > 12, the
number of words to skip is incremented by an additional N12. The start location for data storage
(NS) for logical units < 13 = NFRST + NSKP. The start location for data storage for logical units ≥ 1
is computed as follows:

\[ NS = NSKP + (2 \times \left(\frac{(NFRST - 1)/NWRS}{LU - 13}\right) \times NWRS + 1. \]

The ending location for data storage (NE) = NS + NWRS. Contiguous data elements beginning
with the first word of array A are stored in array C in locations NS through NE.

ENTRY ORD
ENTRY ORD initializes the virtual disk from tape. The array BIG (equivalenced to C) is read
from the tape drive logical unit number LO.

ENTRY OWRT
ENTRY OWRT saves the virtual disk data to tape. Array BIG (equivalenced to C) is stored on
the tape drive logical unit number LO.

ENTRY OCLOSE
In core-contained mode, ENTRY OCLOSE does not perform any function but serves as a
placeholder for future conversion to actual disk I/O.

ENTRY OFIND
In core-contained mode, ENTRY OFIND does not perform any function but serves as a place-
holder for future conversion to actual disk I/O.

3.5 CSC Next Timestep

The CSC Next Timestep is used after each timestep run of CSCs Ice and Ocean to update
timestep information. Its CSU SWAP calculates a new date-time group, new currents, and new
mixed-layer temperatures for the next pass through the ice and ocean models.

3.5.1 CSU SWAP

CSU SWAP meets the requirements of CSC Next Timestep for the next timestep. It also swaps
positions of currents, temperatures and salinities and calculates a new mixed-layer temperature.

3.5.1.1 CSU SWAP Design Specification/Constraint

There are no known constraints.

3.5.1.2 CSU SWAP Design

Input Data Elements:
/COX2/

FW1  heat above the freezing temperature
GICE ice growth rate computed by the ice model
TM1  previous timestep mixed-layer temperature
TM2  previous two timestep mixed-layer temperature
/CURNTS/
T0 present timestep of temperature and salinity
T1 previous timestep of temperature and salinity
T2 two timesteps ago of temperature and salinity
U0 present timestep of x component of ocean current
U1 previous timestep of x component of ocean current
V0 present timestep of y component of ocean current
V1 previous timestep of y component of ocean current

/FULLWD/
NLAST final timestep to compute on this run of model

/OCEANS/
FW oceanic heat flux

/TSTEP/
IDTG date-time group; read from standard input (YMMDDHH)
MDY day calculated from date-time group
MHR hour calculated from date-time group
MM month calculated from date-time group
MYR year calculated from date-time group

Output Data Elements:

/COX2/
GICE ice growth rate computed by the ice model
TFRZ temperature at freezing point
TM1 previous timestep mixed-layer temperature
TM2 previous two timestep mixed-layer temperature

/CURNTS/
T1 previous timestep of temperature and salinity
T2 two timesteps ago of temperature and salinity
U1 previous timestep of x component of ocean current
V1 previous timestep of y component of ocean current

/FULLWD/
NFIRST restart indicator; = 1 to start from scratch; = 0 to restart from data
NLAST final timestep to compute on this run of model
NNERGY 10 timesteps past NLAST

/OCEANS/
FW oceanic heat flux

/RFOR2/
GWATX x component of the ocean current
GWATY y component of the ocean current

/TSTEP/
IDTG date-time group; read from standard input (YMMDDHH)
MDY day calculated from date-time group
MHR hour calculated from date-time group
MM month calculated from date-time group
MYR year calculated from date-time group
Local Data Elements:

- CP: constant of water heat capacity, real
- DELTAT: timestep in seconds, real
- EXCHNG: number of times to exchange ice-ocean data, real
- I: index counter, integer
- IHR: timestep in hours, integer
- IMON: index counter for obtaining current month, integer
- J: index counter, integer
- KK: index counter, integer
- RLATNT: constant of water latent heat, real
- TDAY: timestep to exchange ice-ocean data, real
- ZMIX: depth of mixed layer (30 m), real

Logic Flow:

The next timesteps’ date-time group, year, month, day, and hour values are calculated. The timestep counters are updated. Previous timestep temperature, salinity, and current fields are moved one timesteps backward. Present timestep fields are moved to the previous timestep fields. New seawater freezing temperatures are calculated according to salinity as \(-54.4 \times \text{salinity}\). Ice thickness, growth rate of open water, the total ice thickness growth rate, and the heat above the freezing temperature from the present timestep are swapped with the previous timestep. If the growth rate is greater than that of the heat above freezing temperature, and the heat above freezing is > 0 the growth rate is set to 0 (no more cooling). A new heat flux is calculated and its value is divided by 118.

Algorithms:

The new oceanic heat flux is calculated using:

\[
Q = c_p \ z_m \ (T_{m_i} - T_{m_{i-1}})/t_d/n_t + G L_w,
\]

where:
- \(c_p\) is the constant of water heat capacity,
- \(z_m\) is the depth of the mixed layer,
- \(T_{m_i}\) is the mixed-layer temperature from the previous timestep,
- \(T_{m_{i-1}}\) is the mixed-layer temperature from two timesteps ago,
- \(t_d/n_t\) is the timestep on which to exchange data (\(t_d\) is the number of seconds in a day and \(n_t\) is the number of times to exchange data),
- \(G\) is the ice growth rate, and
- \(L_w\) is the constant of water latent heat.

3.6 CSC Output

The CSC Output is the last CSC executed in PIPS2.0. It is responsible for writing the data file containing the ice model and ocean model restart data. These restart values are used by PIPS2.0 for the next model run.

3.6.1 CSU SAV_ICE

The purpose of the CSU SAV_ICE is to write the ice model restart fields to a file.

3.6.1.1 CSU SAV_ICE Design Specification/Constraint

The ice model restart data are written to a file connected to unit 33.
3.6.1.2 CSU SAV\_ICE Design

Input Data Elements:

```
/RSTRT/
AREA1  fraction of the grid cell covered by thick ice
HEFF   mean ice thickness per grid cell
TICE   mixed-layer temperature for open water or ice temperature for ice cover
UICE   x component of the ice drift
UICEC  intermediate x component of the ice drift
VICE   y component of the ice drift
VICEC  intermediate y component of the ice drift
```

Output Data Elements:

```
UICE   x component of the ice drift
UICEC  intermediate x component of the ice drift
VICE   y component of the ice drift
VICEC  intermediate y component of the ice drift
HEFF   mean ice thickness per grid cell
AREA1  fraction of the grid cell covered by thick ice
TICE   mixed-layer temperature for open water or ice temperature for ice cover
```

Parameters:

stored in file ice.par

The ice.par parameters are described in Sec. 3.1.1.2.

Data element design information is provided in Sec. 4.0.

Logic Flow:

The SAV\_ICE CSU is called by CSU DRIVER.

All restart fields are written to a file connected to unit 33. These restart fields include x and y components of the ice drift (UICE and VICE), intermediate x and y components of the ice drift (UICEC and VICEC), mean ice thickness per grid cell (HEFF), fraction of the grid cell covered by thick ice (AREA1), and mixed-layer temperature for open water or ice temperature for ice cover (TICE).

Local data files or database:

The restart data are written to the 92\<mmdd\>\res file, where \<mmdd\> is the month and day of the current day. This file is a permanent, binary file that contains the x and y components of the ice drift, the intermediate x and y components of the ice drift, the mean ice thickness per grid cell, the fraction of the cell covered by thick ice, and the mixed-layer temperature in the case of open water or the ice temperature in the case of an ice cover. The data in this file are used to restart a run.

3.6.2 CSU SAV\_OCN

The purpose of the CSU SAV\_OCN is to write the ocean model restart fields to a file.

3.6.2.1 CSU SAV\_OCN Design Specification/Constraint

The ocean model restart data are written to a file connected to unit 30.
3.6.2.2 CSU SAV_OCN Design

Input Data Elements:

/COX2/
FW1 heat above the freezing temperature
GICE ice growth rate computed by the ice model
SHICE total ice thickness

/CURNTS/
T0 present timestep of temperature and salinity
T1 previous timestep of temperature and salinity
U0 present timestep of x component of ocean current
U1 previous timestep of x component of ocean current
V0 present timestep of y component of ocean current
V1 previous timestep of y component of ocean current

/FULLWD/
ITT timestep counter

Output Data Elements:

ITT timestep counter
U0 present timestep of x component of ocean current
V0 present timestep of y component of ocean current
T0 present timestep of temperature and salinity
U1 previous timestep of x component of ocean current
V1 previous timestep of y component of ocean current
T1 previous timestep of temperature and salinity

Parameters:
stored in file ice.par
The ice.par parameters are described in Sec. 3.1.1.2.

Local Data Elements:

I index counter
J index counter
M index counter

Logic Flow:

The CSU SAV_OCN is called by CSU DRIVER.

All restart fields are written to a file connected to unit 30. The restart fields include the ice growth rate, present and previous timesteps of temperature and salinity, and the present and previous timesteps of the x and y components of ocean current.

Local data files or database:
The restart data are written to the for010_<mmdd>.dat (unit 30) file, where <mmdd> is the month and day of the current day. This file is a permanent, binary file. The data in this file is used to restart the next model run.

4.0 PIPS2.0 DATA TABLES

Data Tables are presented for the two higher level CSCs: CSC Compute Ice and CSC Compute Ocean.
4.1 Ice Data Table

This section describes the global data elements of CSC Compute Ice.

<table>
<thead>
<tr>
<th>NAME</th>
<th>DATA TYPE</th>
<th>DESCRIPTION</th>
<th>CSU DEFINED IN</th>
<th>CSU USED IN</th>
<th>COMMON BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>A22</td>
<td>real</td>
<td>minimal compactness allowed defined by data statement; (0.15)</td>
<td>ICEMDL, HEAT</td>
<td>GROWTH, HEAT, ICEMDL</td>
<td>local variable</td>
</tr>
<tr>
<td>AMASS</td>
<td>real</td>
<td>ice mass per grid area defined by data statement; array with dimensions (MT,MT)</td>
<td>FORM</td>
<td>FORM, ICEMDL, RELAX</td>
<td>local variable</td>
</tr>
<tr>
<td>AREA1</td>
<td>real</td>
<td>fraction of grid cell covered by ice; read from restart data file or initialized to constant initial condition; array with dimensions (MT,MT,3)</td>
<td>RST_ICE</td>
<td>DRIVER, FORM, GROWTH, HEAT, ICEMDL, RST_ICE, SAV_ICE, SSMI, STRESSUP, TRACER</td>
<td>RSTRT</td>
</tr>
<tr>
<td>CFO</td>
<td>real</td>
<td>heat above freezing in terms of ice thickness growth rate or ice thickness growth rate in open water; array with dimensions (MT,MT)</td>
<td>HEAT</td>
<td>HEAT</td>
<td>COX2</td>
</tr>
<tr>
<td>DELTAT</td>
<td>real</td>
<td>timestep (s)</td>
<td>ICEMDL</td>
<td>ADVECT, BUDGET, GROWTH, ICEMDL, RELAX, SWAP</td>
<td>STEP</td>
</tr>
<tr>
<td>DELTAX</td>
<td>real</td>
<td>x (longitude) grid spacing (deg)</td>
<td>ICEMDL, SWAP</td>
<td>ADVECT, DIFFUS, ICEMDL, PLAST, SWAP</td>
<td>STEP</td>
</tr>
<tr>
<td>DELTAY</td>
<td>real</td>
<td>y (latitude) grid spacing (deg)</td>
<td>ICEMDL, SWAP</td>
<td>ADVECT, DIFFUS, ICEMDL, PLAST, RELAX, SWAP</td>
<td>STEP</td>
</tr>
<tr>
<td>DELTT</td>
<td>real</td>
<td>timestep based on type of timestep differencing (h)</td>
<td>ICEMDL</td>
<td>ADVECT, DIFFUS, ICEMDL</td>
<td>local variable</td>
</tr>
<tr>
<td>DELTXA</td>
<td>real</td>
<td>x (longitude) grid spacing for thermodynamic fields (m); array with dimensions (MT)</td>
<td>ICEMDL</td>
<td>ADVECT, DIFFUS, PLAST, ICEMDL</td>
<td>STEPS</td>
</tr>
<tr>
<td>DELTXU</td>
<td>real</td>
<td>x (longitude) grid spacing for velocity fields (m); array with dimensions (MTM1)</td>
<td>ICEMDL</td>
<td>FORM, GEOWIND, ICEMDL, RELAX</td>
<td>STEPS</td>
</tr>
<tr>
<td>NAME</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
<td>CSU DEFINED IN</td>
<td>CSU USED IN</td>
<td>COMMON BLOCK</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>-----------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>DELTYA</td>
<td>real</td>
<td>y (latitude) grid spacing for thermodynamic fields (m)</td>
<td>ICEMDL</td>
<td>ADEVC, DIFFUS, ICEMDL, PLAST</td>
<td>STEPSP</td>
</tr>
<tr>
<td>DELTYU</td>
<td>real</td>
<td>y (latitude) grid spacing for velocity fields (m)</td>
<td>ICEMDL</td>
<td>FORM, GEOWIND, ICEMDL, RELAX</td>
<td>STEPSP</td>
</tr>
<tr>
<td>DIFF1</td>
<td>real</td>
<td>harmonic diffusion constant</td>
<td>ICEMDL</td>
<td>ADEVC, DIFFUS, ICEMDL</td>
<td></td>
</tr>
<tr>
<td>DIFF3</td>
<td>real</td>
<td>harmonic diffusion constant; array with dimensions (JMT)</td>
<td>ADEVC</td>
<td>ADEVC, DIFFUS</td>
<td>DIFFU3</td>
</tr>
<tr>
<td>DRAGA</td>
<td>real</td>
<td>asymmetric water drag plus the Coriolis parameter; array with dimensions (JMT,JMT)</td>
<td>FORM</td>
<td>RELAX, ICEMDL</td>
<td>local variable</td>
</tr>
<tr>
<td>DRAGS</td>
<td>real</td>
<td>symmetric water drag; array with dimensions (JMT,JMT)</td>
<td>FORM</td>
<td>RELAX, ICEMDL</td>
<td>local variable</td>
</tr>
<tr>
<td>ERROR</td>
<td>real</td>
<td>maximum error allowed in the relation scheme defined by data statement (0.0000001)</td>
<td>ICEMDL</td>
<td>ICEMDL, RELAX</td>
<td>local variable</td>
</tr>
<tr>
<td>ES</td>
<td>real</td>
<td>NOGAPS atmospheric forcing — surface vapor pressure; array with dimensions (JMT,JMT)</td>
<td>RFORCE</td>
<td>HEAT, RFORCE</td>
<td>RFOR</td>
</tr>
<tr>
<td>ES1</td>
<td>real</td>
<td>NOGAPS atmospheric forcing — sensible heat flux; array with dimensions (JMT,JMT)</td>
<td>RFORCE</td>
<td>HEAT, RFORCE</td>
<td>RFOR</td>
</tr>
<tr>
<td>ETA</td>
<td>real</td>
<td>nonlinear shear viscosity; array with dimensions (JMT,JMT)</td>
<td>PLAST</td>
<td>FORM, ICEMDL, RELAX</td>
<td>local variable</td>
</tr>
<tr>
<td>FCORSP</td>
<td>real</td>
<td>array of sine of latitude positions of each gridpoint; array with dimensions (JMT1,JMT1)</td>
<td>DRIVER</td>
<td>DRIVER, FORM, GEOWIND, OCEAN</td>
<td>CORSP</td>
</tr>
<tr>
<td>FHEFF</td>
<td>real</td>
<td>total ice growth rate of thick ice; array with dimensions (JMT,JMT)</td>
<td>HEAT</td>
<td>GROWTH, HEAT, ICEMDL,</td>
<td>GROW</td>
</tr>
<tr>
<td>FO</td>
<td>real</td>
<td>growth rate of thin ice; array with dimensions (NX1,NY1)</td>
<td>HEAT</td>
<td>GROWTH, HEAT, ICEMDL,</td>
<td>local variable</td>
</tr>
<tr>
<td>FORCEX</td>
<td>real</td>
<td>x component of forcing due to the ocean currents plus the ice pressure gradient; array with dimensions (JMT1,JMT1)</td>
<td>FORM</td>
<td>RELAX, FORM</td>
<td>FORCE</td>
</tr>
<tr>
<td>FORCEY</td>
<td>real</td>
<td>x component of forcing due to the ocean currents plus the ice pressure gradient; array with dimensions (JMT1,JMT1)</td>
<td>FORM</td>
<td>RELAX, FORM</td>
<td>FORCE</td>
</tr>
<tr>
<td>NAME</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
<td>CSU DEFINED IN</td>
<td>CSU USED IN</td>
<td>COMMON BLOCK</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>-------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>FSH</td>
<td>real</td>
<td>atmospheric forcing — solar radiation; array with dimensions (/MTP1,/MTP1)</td>
<td>RFORCE</td>
<td>BUDGET, HEAT, RFORCE</td>
<td>RAD</td>
</tr>
<tr>
<td>FW</td>
<td>real</td>
<td>oceanic heat flux; array with dimensions (/MT,/MT)</td>
<td>RST_OCN</td>
<td>BUDGET, HEAT, OCEAN, RST_OCN, SWAP</td>
<td>OCEANS</td>
</tr>
<tr>
<td>FW1</td>
<td>real</td>
<td>heat above the freezing temperature; array with dimensions (/MT,/MT)</td>
<td>HEAT, RST_OCN</td>
<td>HEAT, RST_OCN, SAV_OCN, TRACER</td>
<td>COX2</td>
</tr>
<tr>
<td>FW2</td>
<td>real</td>
<td>oceanic heat flux (watts/m²); array with dimensions (/MT,/MT)</td>
<td>BUDGET</td>
<td>BUDGET</td>
<td>COX2</td>
</tr>
<tr>
<td>GAIRX</td>
<td>real</td>
<td>x component of the geostrophic wind; array with dimensions (/MTP1,/MTP1)</td>
<td>RFORCE</td>
<td>DRIVER, FORM, HEAT, ICEMDL, RFORCE, STRESSUP</td>
<td>RFOR2</td>
</tr>
<tr>
<td>Gairy</td>
<td>real</td>
<td>y component of the geostrophic wind; array with dimensions (/MTP1,/MTP1)</td>
<td>RFORCE</td>
<td>DRIVER, FORM, HEAT, ICEMDL, RFORCE, STRESSUP</td>
<td>RFOR2</td>
</tr>
<tr>
<td>GAREA</td>
<td>real</td>
<td>change of compactness due to freezing or melting; array with dimensions (/MTP1,/MTP1)</td>
<td>GROWTH</td>
<td>ICEMDL</td>
<td>local variable</td>
</tr>
<tr>
<td>GICE</td>
<td>real</td>
<td>ice thickness growth rate of open water; array with dimensions (/MT,/MT)</td>
<td>HEAT, RST_OCN</td>
<td>HEAT, RST_OCN, SAV_OCN, SWAP, TRACER</td>
<td>COX2</td>
</tr>
<tr>
<td>GWATX</td>
<td>real</td>
<td>x component of ocean current (m/s); array with dimensions (/MTM1,/MTM1)</td>
<td>RST_OCN, SWAP</td>
<td>DRIVER, FORM, ICEMDL, RST_OCN, SWAP</td>
<td>RFOR2</td>
</tr>
<tr>
<td>GWATY</td>
<td>real</td>
<td>y component of ocean current (m/s); array with dimensions (/MTM1,/MTM1)</td>
<td>RST_OCN, SWAP</td>
<td>DRIVER, FORM, ICEMDL, RST_OCN, SWAP</td>
<td>RFOR2</td>
</tr>
<tr>
<td>HCORR</td>
<td>real</td>
<td>additional ice to be melted for mixed-layer balance; array with dimensions (/MT,/MT)</td>
<td>GROWTH</td>
<td>ICEMDL</td>
<td>local variable</td>
</tr>
<tr>
<td>HDIFF1</td>
<td>real</td>
<td>net rate of total open water growth; array with dimensions (/MT,/MT)</td>
<td>HEAT</td>
<td>GROWTH, ICEMDL</td>
<td>local variable</td>
</tr>
<tr>
<td>NAME</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
<td>CSU DEFINED IN</td>
<td>CSU USED IN</td>
<td>COMMON BLOCK</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------</td>
<td>------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>HEFF</td>
<td>real</td>
<td>mean ice thickness per grid cell; read from restart data file or initialized to constant initial condition; array with dimensions (/MT,/MT,3)</td>
<td>RST_ICE</td>
<td>ADVECT, DIFFUS, DRIVER, FORM, GROWTH, ICEMDL, HEAT, RNEGT, RST_ICE, RST_OCN, SAV_ICE, SSML, STRESSUP, XSUM</td>
<td>RSTRRT</td>
</tr>
<tr>
<td>HEFFM</td>
<td>real</td>
<td>land/sea mask for thermodynamic variables; array with dimensions (/MT,/MT)</td>
<td>BNDRY</td>
<td>ADVECT, BNDRY, DIFFUS, DRIVER, FORM, GROWTH, ICEMDL, PLAST</td>
<td>MASK</td>
</tr>
<tr>
<td>HO</td>
<td>real</td>
<td>demarcation between thick and thin ice defined by data statement; (0.5 m)</td>
<td>ICEMDL</td>
<td>ICEMDL, GROWTH</td>
<td>local variable</td>
</tr>
<tr>
<td>IDTG</td>
<td>integer</td>
<td>date-time group; read from standard input (YYMMDDHH)</td>
<td>ICEMDL</td>
<td>DAYNUM, ICEMDL, SWAP</td>
<td>TSTEP</td>
</tr>
<tr>
<td>IMT</td>
<td>integer</td>
<td>total number of T grid boxes zonally (360)</td>
<td></td>
<td></td>
<td>local variable</td>
</tr>
<tr>
<td>IMTM1</td>
<td>integer</td>
<td>total number of T grid boxes zonally – 1 (359)</td>
<td></td>
<td></td>
<td>local variable</td>
</tr>
<tr>
<td>IMTP1</td>
<td>integer</td>
<td>total number of T grid boxes zonally + 1 (361)</td>
<td></td>
<td></td>
<td>local variable</td>
</tr>
<tr>
<td>IRSTRT</td>
<td>integer</td>
<td>if 0 restarts from previous run; otherwise, restarts from constant conditions; read from standard input</td>
<td>DRIVER</td>
<td>DRIVER</td>
<td>TSTEP</td>
</tr>
<tr>
<td>ITSTEP</td>
<td>integer</td>
<td>number of timesteps for run; read from standard input</td>
<td>DRIVER</td>
<td>DRIVER, ICEMDL</td>
<td>TSTEP</td>
</tr>
<tr>
<td>JMT</td>
<td>integer</td>
<td>total number of T grid boxes meridionally (360)</td>
<td></td>
<td></td>
<td>local variable</td>
</tr>
<tr>
<td>JMTM1</td>
<td>integer</td>
<td>total number of T grid boxes meridionally – 1 (359)</td>
<td></td>
<td></td>
<td>local variable</td>
</tr>
<tr>
<td>JMPI</td>
<td>integer</td>
<td>total number of T grid boxes meridionally + 1 (361)</td>
<td></td>
<td></td>
<td>local variable</td>
</tr>
<tr>
<td>LAD</td>
<td>integer</td>
<td>type of timestepping defined by data statement (2)</td>
<td>ICEMDL</td>
<td>ADVECT, ICEMDL</td>
<td>local variable</td>
</tr>
<tr>
<td>NAME</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
<td>CSU DEFINED IN</td>
<td>CSU USED IN</td>
<td>COMMON BLOCK</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>---------------------------------------------------------------</td>
<td>----------------</td>
<td>-----------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>MDY</td>
<td>integer</td>
<td>day calculated from date-time group</td>
<td>DAYNUM</td>
<td>DAYNUM, DRIVER, ICEMDL, SWAP</td>
<td>TSTEP</td>
</tr>
<tr>
<td>MHR</td>
<td>integer</td>
<td>hour calculated from date-time group</td>
<td>DAYNUM</td>
<td>DAYNUM, DRIVER, ICEMDL, SWAP</td>
<td>TSTEP</td>
</tr>
<tr>
<td>MIDY</td>
<td>integer</td>
<td>defined by parameter statement (160)</td>
<td>ICEMDL, ADVECT, OCEAN, RELAX</td>
<td>ADVECT, ICEMDL, OCEAN, RELAX</td>
<td>local variable</td>
</tr>
<tr>
<td>MM</td>
<td>integer</td>
<td>month calculated from date-time group</td>
<td>DAYNUM</td>
<td>DAYNUM, DRIVER, ICEMDL, SWAP</td>
<td>TSTEP</td>
</tr>
<tr>
<td>MYR</td>
<td>integer</td>
<td>year calculated from date-time group</td>
<td>DAYNUM</td>
<td>DAYNUM, DRIVER, ICEMDL, SWAP</td>
<td>TSTEP</td>
</tr>
<tr>
<td>NXM1</td>
<td>integer</td>
<td>/MT-2; defined by parameter statement (358)</td>
<td>ADVECT</td>
<td>ADVECT</td>
<td>local variable</td>
</tr>
<tr>
<td>NYM1</td>
<td>integer</td>
<td>/MT-2; defined by parameter statement (358)</td>
<td>ADVECT</td>
<td>ADVECT</td>
<td>local variable</td>
</tr>
<tr>
<td>OUT</td>
<td>real</td>
<td>land/sea mask including outflow conditions for thermodynamic variables; array with dimensions (IMT, JMT)</td>
<td>BNDRY</td>
<td>BNDRY, DRIVER, FORM, GROWTH, ICEMDL</td>
<td>MASK</td>
</tr>
<tr>
<td>PLTSTP</td>
<td>integer</td>
<td>interval in timesteps at which to plot results; read from standard input</td>
<td>DRIVER</td>
<td>DRIVER</td>
<td>TSTEP</td>
</tr>
<tr>
<td>PRESS</td>
<td>real</td>
<td>ice strength; array with dimensions (IMT, JMT)</td>
<td>FORM</td>
<td>PLAST, FORM</td>
<td>PRESS</td>
</tr>
<tr>
<td>PRTSTP</td>
<td>integer</td>
<td>interval in timesteps at which to write results; read from standard input</td>
<td>DRIVER</td>
<td>DRIVER, ICEMDL</td>
<td>TSTEP</td>
</tr>
<tr>
<td>PS</td>
<td>real</td>
<td>NOGAPS atmospheric forcing—surface air pressure; array with dimensions (JMTP1, JMTP1)</td>
<td>RFORCE</td>
<td>HEAT, GEOWIND, RFORCE</td>
<td>RFOR</td>
</tr>
<tr>
<td>PS1</td>
<td>real</td>
<td>NOGAPS atmospheric forcing—total heat flux; array with dimensions (JMTP1, JMTP1)</td>
<td>RFORCE</td>
<td>HEAT, RFORCE</td>
<td>RFOR</td>
</tr>
<tr>
<td>RADIAN</td>
<td>real</td>
<td>deg per radian defined by parameter statement (57.29578°)</td>
<td>ADVECT, DRIVER, OCEAN, RELAX, ICEMDL</td>
<td>ADVECT, DRIVER, OCEAN, RELAX, ICEMDL</td>
<td>local variable</td>
</tr>
<tr>
<td>NAME</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
<td>CSU DEFINED IN</td>
<td>CSU USED IN</td>
<td>COMMON BLOCK</td>
</tr>
<tr>
<td>-------</td>
<td>-----------</td>
<td>--------------------------------------------------</td>
<td>----------------------</td>
<td>-------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>RADIUS</td>
<td>real</td>
<td>earth's radius defined by parameter statement (6370.0 * 10^3 m)</td>
<td>ADVECT, CLINIC, ICEMDL, RELAX</td>
<td>ADVECT, CLINIC, ICEMDL, RELAX</td>
<td>local variable</td>
</tr>
<tr>
<td>SHICE</td>
<td>real</td>
<td>total ice thickness; array with dimensions (/MT,/MT)</td>
<td>HEAT, RST_OCN</td>
<td>HEAT, RST_OCN, SAV_OCN, TRACER</td>
<td>COX2</td>
</tr>
</tbody>
</table>
| SNRT  | real      | • Jan, Feb, Mar, or Apr (3.215 x 10^9)  
• May (19.29 x 10^9)  
• Jun, Jul, or Aug 1–19 (0.0)  
• Aug 20–31 (49.603 x 10^9)  
• Sep or Oct (49.603 x 10^9)  
• Nov or Dec (3.215 x 10^9) | ICEMDL | BUDGET, ICEMDL | SNOW |
<p>| TAIR  | real      | NOGAPS atmospheric forcing—surface air temperature; array with dimensions (/MTIP1,/MT1) | RFORCE | BUDGET, HEAT, RFORCE | RFOR |
| TFRZ  | real      | temperature at the freezing point; array with dimensions (/MT,/MT) | RST_OCN, SWAP | HEAT, RST_OCN, SSMI, SWAP | COX2 |
| THETA | real      | indicates backward timestep                      | ICEMDL | ICEMDL, RELAX | local variable |
| TICE  | real      | mixed-layer temperature for open water or ice temperature for ice cover (Kelvin); read from restart data file or initialized to constant initial condition; array with dimensions (/MT,/MT) | RST_OCE | BUDGET, DRIVER, RST_ICE, SAV_ICE | RSTRT |
| TM1   | real      | mixed-layer temperature (Kelvin) from previous timestep; array with dimensions (/MT,/MT) | RST_OCN, SSMI | HEAT, SSMI, RST_OCN, SWAP | COX2 |
| TM2   | real      | mixed-layer temperature from two timesteps ago (Kelvin); array with dimensions (/MT,/MT) | RST_OCN, SSMI | RST_OCN, SSMI, SWAP | COX2 |
| TEMP  | real      | ocean salinity and temperature (°C) from previous timestep; array with dimensions (/MT,2) | RST_OCN | RST_OCN | COX2 |
| UICE  | real      | x component of ice drift; array with dimensions (/MTM1,/MTM1,3) | ICEMDL or RST_ICE | DRIVER, FORM, ICEMDL, PLAST, RELAX, RST_ICE, SAV_ICE, STRESSUP | RSTRT |</p>
<table>
<thead>
<tr>
<th>NAME</th>
<th>DATA TYPE</th>
<th>DESCRIPTION</th>
<th>CSU DEFINED IN</th>
<th>CSU USED IN</th>
<th>COMMON BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>UICEC</td>
<td>real</td>
<td>intermediate x component of the ice drift; read from restart data file or initialized to constant initial condition; array with dimensions (/MTM1,/MTM1,3)</td>
<td>ICEMDL or RST_ICE</td>
<td>ADVECT, DRIVER, ICEMDL, RELAX, RST_ICE, SAV_ICE</td>
<td>RSTRT</td>
</tr>
<tr>
<td>UTEMP</td>
<td>real</td>
<td>x component of ocean current from previous timestep; array with dimension (MT)</td>
<td>RST_OCN</td>
<td>RST_OCN</td>
<td>COX2</td>
</tr>
<tr>
<td>UVM</td>
<td>real</td>
<td>land/sea mask for velocity variables; array with dimensions (/MTM1,/MTM1)</td>
<td>BNDRY</td>
<td>BNDRY, ICEMDL, RELAX</td>
<td>MASK</td>
</tr>
<tr>
<td>VICE</td>
<td>real</td>
<td>y component of ice drift; array with dimensions (/MTM1,/MTM1,3)</td>
<td>ICEMDL or RST_ICE</td>
<td>DRIVER, FORM, ICEMDL, PLAST, RELAX, RST_ICE, SAV_ICE, STRESSUP</td>
<td>RSTRT</td>
</tr>
<tr>
<td>VICEC</td>
<td>real</td>
<td>intermediate y component of the ice drift; read from restart data file or initialized to constant initial condition; array with dimensions (/MTM1,/MTM1,3)</td>
<td>ICEMDL or RST_ICE</td>
<td>ADVECT, DRIVER, ICEMDL, RELAX, RST_ICE, SAV_ICE</td>
<td>RSTRT</td>
</tr>
<tr>
<td>VTEMP</td>
<td>real</td>
<td>y component of ocean current from previous timestep; array with dimension (MT)</td>
<td>RST_OCN</td>
<td>RST_OCN</td>
<td>COX2</td>
</tr>
<tr>
<td>ZETA</td>
<td>real</td>
<td>nonlinear bulk viscosity; array with dimensions (MT,M)</td>
<td>PLAST</td>
<td>FORM, ICEMDL, RELAX</td>
<td>local variable</td>
</tr>
</tbody>
</table>
4.2 Ocean Data Table

This section describes the global data elements of CSC Compute Ocean.

<table>
<thead>
<tr>
<th>DATA ELEMENT</th>
<th>DATA TYPE</th>
<th>DESCRIPTION</th>
<th>CSU DEFINED IN</th>
<th>CSU USED IN</th>
<th>COMMON BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABT</td>
<td>real</td>
<td>temporary array for printing topography map</td>
<td>OCEAN</td>
<td>OCEAN</td>
<td>local variable</td>
</tr>
<tr>
<td>ACOR</td>
<td>real</td>
<td>if $= 0$, treat the Coriolis term explicitly if $&gt; 0$, treat the Coriolis term implicitly with forward component weighted by ACOR, past component by $1 - ACOR$</td>
<td>OCEAN</td>
<td>CLINIC, RELAX</td>
<td>SCALAR</td>
</tr>
<tr>
<td>ACORF</td>
<td>real</td>
<td>$= ACOR$; read in NAMELIST PARMS</td>
<td>OCEAN</td>
<td>OCEAN</td>
<td>local variable</td>
</tr>
<tr>
<td>AH</td>
<td>real</td>
<td>coefficient of horizontal mixing of $T$</td>
<td>OCEAN</td>
<td>STEP</td>
<td>SCALAR</td>
</tr>
<tr>
<td>AHF</td>
<td>real</td>
<td>$= AH$; read in NAMELIST EDDY</td>
<td>OCEAN</td>
<td>OCEAN</td>
<td>local variable</td>
</tr>
<tr>
<td>AICE</td>
<td>real</td>
<td>ice concentration read from restart file created by the ice model</td>
<td>OCEAN</td>
<td>TRACER</td>
<td>TSTOP</td>
</tr>
<tr>
<td>AKNTRL</td>
<td>real</td>
<td>available for storage of values that need to be saved to the restart file</td>
<td>not used</td>
<td>not used</td>
<td>FULLWD</td>
</tr>
<tr>
<td>AM</td>
<td>real</td>
<td>coefficient of horizontal mixing of $U,V$</td>
<td>OCEAN</td>
<td>CLINIC</td>
<td>SCALAR</td>
</tr>
<tr>
<td>AMF</td>
<td>real</td>
<td>$= AM$; read in NAMELIST EDDY</td>
<td>OCEAN</td>
<td>OCEAN</td>
<td>local variable</td>
</tr>
<tr>
<td>AREA</td>
<td>real</td>
<td>area of the surface of the model basin</td>
<td>OCEAN</td>
<td>FULLWD</td>
<td></td>
</tr>
<tr>
<td>ASH</td>
<td>real</td>
<td>set to 0 for a grid cell that meets the criteria for open water</td>
<td>TRACER</td>
<td>TRACER</td>
<td>local variable</td>
</tr>
<tr>
<td>BBTJ</td>
<td>real</td>
<td>coefficient used in horizontal mixing of $T$</td>
<td>TRACER</td>
<td>TRACER</td>
<td>local variable</td>
</tr>
<tr>
<td>BBUJ</td>
<td>real</td>
<td>coefficient used in horizontal mixing of $U,V$</td>
<td>CLINIC</td>
<td>CLINIC</td>
<td>local variable</td>
</tr>
<tr>
<td>BCON</td>
<td>real</td>
<td>slab incidental data on $N + 1$ slab</td>
<td>STEP</td>
<td>STEP</td>
<td>WORKSP</td>
</tr>
<tr>
<td>BIG</td>
<td>real</td>
<td>virtual disk, stores disk data in core-contained mode</td>
<td>ODAM</td>
<td>ODAM</td>
<td>BG</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
<td>CSU DEFINED IN</td>
<td>CSU USED IN</td>
<td>COMMON BLOCK</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>----------------------------------------------------------</td>
<td>----------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>BLK</td>
<td>character*1</td>
<td>a single blank character</td>
<td>OCEAN</td>
<td>OCEAN</td>
<td>local variable</td>
</tr>
<tr>
<td>BOXVOL</td>
<td>real</td>
<td>volume of one grid box</td>
<td>TRACER</td>
<td>TRACER</td>
<td>local variable</td>
</tr>
<tr>
<td>BUOY</td>
<td>real</td>
<td>energy transfer through buoyancy effects</td>
<td>TRACER, STEP</td>
<td>TRACER</td>
<td>FULLWD</td>
</tr>
<tr>
<td>C</td>
<td>real</td>
<td>coefficients of equation of state</td>
<td>STATE</td>
<td>STATE</td>
<td>local variable</td>
</tr>
<tr>
<td>C2DTSF</td>
<td>real</td>
<td>DTSF × 2</td>
<td>STEP</td>
<td>CLINIC</td>
<td>SCALAR</td>
</tr>
<tr>
<td>C2DTTS</td>
<td>real</td>
<td>DTTS × 2</td>
<td>STEP</td>
<td>TRACER</td>
<td>SCALAR</td>
</tr>
<tr>
<td>C2DTUV</td>
<td>real</td>
<td>DTUV × 2</td>
<td>STEP</td>
<td>CLINIC</td>
<td>SCALAR</td>
</tr>
<tr>
<td>C2DZ</td>
<td>real</td>
<td>DZ × 2</td>
<td>OCEAN</td>
<td>STEP</td>
<td>ONEDIM</td>
</tr>
<tr>
<td>C2DZQ</td>
<td>real</td>
<td>DZQ × 2</td>
<td>STEP</td>
<td>CLINIC, TRACER</td>
<td>WORKSP</td>
</tr>
<tr>
<td>CCTJ</td>
<td>real</td>
<td>coefficient used in horizontal mixing of T</td>
<td>STEP</td>
<td>STEP, TRACER</td>
<td>local variable</td>
</tr>
<tr>
<td>CCUJ</td>
<td>real</td>
<td>coefficient used in horizontal mixing of U,V</td>
<td>CLINIC</td>
<td>CLINIC</td>
<td>local variable</td>
</tr>
<tr>
<td>CD</td>
<td>real</td>
<td>a constant used in CSU CLINIC to compute vertical diffusion of momentum; = 0.0013</td>
<td>CLINIC</td>
<td>CLINIC</td>
<td>local variable</td>
</tr>
<tr>
<td>CFE</td>
<td>real</td>
<td>coefficient of eastern point in LaPlacian star</td>
<td>RELAX</td>
<td>RELAX</td>
<td>WORKSP (R)</td>
</tr>
<tr>
<td>CFN</td>
<td>real</td>
<td>coefficient of northern point in LaPlacian star</td>
<td>RELAX</td>
<td>RELAX</td>
<td>WORKSP (R)</td>
</tr>
<tr>
<td>CFS</td>
<td>real</td>
<td>coefficient of southern point in LaPlacian star</td>
<td>RELAX</td>
<td>RELAX</td>
<td>WORKSP (R)</td>
</tr>
<tr>
<td>CFW</td>
<td>real</td>
<td>coefficient of western point in LaPlacian star</td>
<td>RELAX</td>
<td>RELAX</td>
<td>WORKSP (R)</td>
</tr>
<tr>
<td>CHENG</td>
<td>real</td>
<td>Levitus climatological data of ocean temperature and salinity</td>
<td>OCEAN</td>
<td>TRACER</td>
<td>TSTOP</td>
</tr>
<tr>
<td>CIQ</td>
<td>real</td>
<td>Knudsen formula coefficients with alternating reference levels</td>
<td>STATE</td>
<td>STATE</td>
<td>WORKSP</td>
</tr>
<tr>
<td>COF</td>
<td>real</td>
<td>normalization array in computation of island flow</td>
<td>RELAX</td>
<td>RELAX</td>
<td>WORKSP (R)</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COFIS</td>
<td>real</td>
<td>integral of COF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONTRL</td>
<td>real</td>
<td>NAMELIST data file of program flow controllers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPF</td>
<td>real</td>
<td>normalization factor used in constructing LaPlacian star</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CQ</td>
<td>real</td>
<td>Knudsen formula coefficients</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRIT</td>
<td>real</td>
<td>criterion for convergence of relaxation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRITF</td>
<td>real</td>
<td>= CRIT; read in NAMELIST PARMS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRTP</td>
<td>real</td>
<td>altered value of CRIT for computational efficiency</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS</td>
<td>real</td>
<td>cosine of U,V point latitudes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSR</td>
<td>real</td>
<td>1.0/CS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CST</td>
<td>real</td>
<td>cosine of T point latitudes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSTR</td>
<td>real</td>
<td>1.0/CST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAYSYR</td>
<td>real</td>
<td>number of days per year; used in CSU STEP to print timestep information.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDTJ</td>
<td>real</td>
<td>coefficient used in horizontal mixing of T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDUJ</td>
<td>real</td>
<td>coefficient used in horizontal mixing of U,V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DETMR</td>
<td>real</td>
<td>reciprocal of determinant of the matrix arising from simultaneous equations of the semi-implicit treatment of the Coriolis term</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAG1</td>
<td>real</td>
<td>temporary storage of diagonal differences for computational efficiency</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAG2</td>
<td>character*1</td>
<td>temporary storage of diagonal differences for computational efficiency</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CSU DEFINED IN</th>
<th>CSU USED IN</th>
<th>COMMON BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELAX</td>
<td>RELAX</td>
<td>WORKSP (R)</td>
</tr>
<tr>
<td>OCEAN</td>
<td>OCEAN</td>
<td>local variable</td>
</tr>
<tr>
<td>RELAX</td>
<td>RELAX</td>
<td>WORKSP (R)</td>
</tr>
<tr>
<td>STATE</td>
<td>STATE</td>
<td>WORKSP</td>
</tr>
<tr>
<td>OCEAN</td>
<td>RELAX</td>
<td>SCALAR</td>
</tr>
<tr>
<td>OCEAN</td>
<td>OCEAN</td>
<td>local variable</td>
</tr>
<tr>
<td>RELAX</td>
<td>RELAX</td>
<td>local variable</td>
</tr>
<tr>
<td>OCEAN</td>
<td>CLINIC, TRACER</td>
<td>ONEDIM</td>
</tr>
<tr>
<td>OCEAN</td>
<td>CLINIC</td>
<td>ONEDIM</td>
</tr>
<tr>
<td>OCEAN</td>
<td>CLINIC, TRACER</td>
<td>ONEDIM</td>
</tr>
<tr>
<td>STEP</td>
<td>STEP</td>
<td>local variable</td>
</tr>
<tr>
<td>TRACER</td>
<td>TRACER</td>
<td>local variable</td>
</tr>
<tr>
<td>CLINIC</td>
<td>CLINIC</td>
<td>local variable</td>
</tr>
<tr>
<td>STEP</td>
<td>STEP</td>
<td>local variable</td>
</tr>
<tr>
<td>STEP</td>
<td>STEP</td>
<td>local variable</td>
</tr>
<tr>
<td>DATA ELEMENTS</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>DOT</td>
<td>real</td>
<td>= .?</td>
</tr>
<tr>
<td>DPDX</td>
<td>real</td>
<td>zonal derivative of hydrostatic pressure; used in computing zonal component of internal mode velocity</td>
</tr>
<tr>
<td>DPDY</td>
<td>real</td>
<td>meridional derivative of hydrostatic pressure; used in computing meridional component of internal mode velocity</td>
</tr>
<tr>
<td>DTABS</td>
<td>real</td>
<td>volume average of absolute change of temperature</td>
</tr>
<tr>
<td>DTSF</td>
<td>real</td>
<td>length of timestep on stream function</td>
</tr>
<tr>
<td>DTSFF</td>
<td>real</td>
<td>= DTSF; read in NAMELIST TSTEP</td>
</tr>
<tr>
<td>DTTS</td>
<td>real</td>
<td>length of timestep on T</td>
</tr>
<tr>
<td>DTTSF</td>
<td>real</td>
<td>= DTTS; read in NAMELIST TSTEP</td>
</tr>
<tr>
<td>DTUVID</td>
<td>real</td>
<td>length of timestep on U,V</td>
</tr>
<tr>
<td>DTUVF</td>
<td>real</td>
<td>= DTUV; read in NAMELIST TSTEP</td>
</tr>
<tr>
<td>DXT</td>
<td>real</td>
<td>zonal grid spacing across T boxes (between U,V points)</td>
</tr>
<tr>
<td>DXT2R</td>
<td>real</td>
<td>1.0/(DXT * 2.0)</td>
</tr>
<tr>
<td>DXT4R</td>
<td>real</td>
<td>1.0/(DXT * 4.0)</td>
</tr>
<tr>
<td>DXT4RQ</td>
<td>real</td>
<td>similar to DXT4R but for vectorization</td>
</tr>
<tr>
<td>DXTQ</td>
<td>real</td>
<td>similar to DXT but for vectorization</td>
</tr>
<tr>
<td>DXTR</td>
<td>real</td>
<td>1.0/DXT</td>
</tr>
<tr>
<td>DXU</td>
<td>real</td>
<td>zonal grid spacing across U,V boxes (between T points)</td>
</tr>
<tr>
<td>DXU2R</td>
<td>real</td>
<td>1.0/(DXU * 2)</td>
</tr>
<tr>
<td>DXU2RQ</td>
<td>real</td>
<td>similar to DXU2R but for vectorization</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>DXU4R</td>
<td>real</td>
<td>1.0/(DXU * 4)</td>
</tr>
<tr>
<td>DXUQ</td>
<td>real</td>
<td>similar to DXU but for vectorization</td>
</tr>
<tr>
<td>DXUR</td>
<td>real</td>
<td>1.0/DXU</td>
</tr>
<tr>
<td>DYT</td>
<td>real</td>
<td>meridional grid spacing across T boxes (between U,V points)</td>
</tr>
<tr>
<td>DYT2R</td>
<td>real</td>
<td>1.0/(DYT * 2.0)</td>
</tr>
<tr>
<td>DYT4R</td>
<td>real</td>
<td>1.0/(DYT * 4.0)</td>
</tr>
<tr>
<td>DYTR</td>
<td>real</td>
<td>1.0/DYT</td>
</tr>
<tr>
<td>DYu</td>
<td>real</td>
<td>meridional grid spacing across U,V boxes (between T points)</td>
</tr>
<tr>
<td>DYU2R</td>
<td>real</td>
<td>1.0/(DYU * 2.0)</td>
</tr>
<tr>
<td>DYU4R</td>
<td>real</td>
<td>1.0/(DYU * 4.0)</td>
</tr>
<tr>
<td>DYUR</td>
<td>real</td>
<td>1.0/DYU</td>
</tr>
<tr>
<td>DZ</td>
<td>real</td>
<td>vertical grid spacing down U,V,T boxes (between W points); vertical layer thickness in centimeters</td>
</tr>
<tr>
<td>DZ2R</td>
<td>real</td>
<td>1.0/(DZU * 2.0)</td>
</tr>
<tr>
<td>DZ2RQ</td>
<td>real</td>
<td>similar to DZ2R but for vectorization</td>
</tr>
<tr>
<td>DZZ</td>
<td>real</td>
<td>vertical grid spacing across W boxes (between U,V,T points)</td>
</tr>
<tr>
<td>DZZ2R</td>
<td>real</td>
<td>1.0/(DZZ * 2.0)</td>
</tr>
<tr>
<td>DZZ2RQ</td>
<td>real</td>
<td>similar to DZZ2R but for vectorization</td>
</tr>
<tr>
<td>DZZQ</td>
<td>logical</td>
<td>similar to DZZ but for vectorization</td>
</tr>
<tr>
<td>EB</td>
<td>namelist</td>
<td>if true, Euler backward step for time mixing; if false, forward timestep for time mixing</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>EDDY</td>
<td>real</td>
<td>NAMELIST data file of eddy mixing coefficients</td>
</tr>
<tr>
<td>EEH</td>
<td>real</td>
<td>upper vertical mixing coefficient of T</td>
</tr>
<tr>
<td>EEHQ</td>
<td>real</td>
<td>similar to EEH but for vectorization</td>
</tr>
<tr>
<td>EEM</td>
<td>real</td>
<td>upper vertical mixing coefficient of U,V</td>
</tr>
<tr>
<td>EEMQ</td>
<td>real</td>
<td>similar to EEM but for vectorization</td>
</tr>
<tr>
<td>EKTOT</td>
<td>real</td>
<td>total kinetic energy normalized by volume</td>
</tr>
<tr>
<td>ENGEXT</td>
<td>real</td>
<td>accumulators of rates of change of kinetic energy of external mode</td>
</tr>
<tr>
<td>ENGINT</td>
<td>real</td>
<td>accumulators of rates of change of kinetic energy of internal mode</td>
</tr>
<tr>
<td>ENGTMP</td>
<td>real</td>
<td>temporary accumulator of change of kinetic energy of external mode</td>
</tr>
<tr>
<td>FFH</td>
<td>real</td>
<td>lower vertical mixing coefficient of T</td>
</tr>
<tr>
<td>FFHQ</td>
<td>real</td>
<td>similar to FFH but for vectorization</td>
</tr>
<tr>
<td>FFM</td>
<td>real</td>
<td>lower vertical mixing coefficient of U,V</td>
</tr>
<tr>
<td>FFMQ</td>
<td>real</td>
<td>similar to FFM but for vectorization</td>
</tr>
<tr>
<td>FINS</td>
<td>real</td>
<td>floating point array to read in start and end indices</td>
</tr>
<tr>
<td>FKMP</td>
<td>real</td>
<td>number of vertical levels of ocean at T points</td>
</tr>
<tr>
<td>FKMQ</td>
<td>real</td>
<td>number of vertical levels of ocean at U,V points</td>
</tr>
<tr>
<td>FKMT</td>
<td>real</td>
<td>number of vertical levels of ocean at T points</td>
</tr>
<tr>
<td>FKMTM</td>
<td>real</td>
<td>FKMT at row $J - 1$</td>
</tr>
<tr>
<td>FKMTP</td>
<td>real</td>
<td>FKMT at row $J + 1$</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FKMU</td>
<td>real</td>
<td>number of vertical levels of ocean at U,V points</td>
</tr>
<tr>
<td>FKMUM</td>
<td>real</td>
<td>FKMU at row J - 1</td>
</tr>
<tr>
<td>FKMUP</td>
<td>real</td>
<td>FKMU at row J + 1</td>
</tr>
<tr>
<td>FKMZ</td>
<td>real</td>
<td>number of vertical levels of ocean at interior (non-land neighboring) T points</td>
</tr>
<tr>
<td>FKPH</td>
<td>real</td>
<td>coefficient of vertical mixing of T</td>
</tr>
<tr>
<td>FKPHF</td>
<td>real</td>
<td>= FKPH; read in NAMELIST EDDY</td>
</tr>
<tr>
<td>FKPM</td>
<td>real</td>
<td>coefficient of vertical mixing of U,V</td>
</tr>
<tr>
<td>FKPMF</td>
<td>real</td>
<td>= FKPM; read in NAMELIST EDDY</td>
</tr>
<tr>
<td>FM</td>
<td>real</td>
<td>masking array for T points; 0 over land and 1 over ocean</td>
</tr>
<tr>
<td>FMM</td>
<td>real</td>
<td>FM at row J - 1</td>
</tr>
<tr>
<td>FMP</td>
<td>real</td>
<td>FM at row J + 1</td>
</tr>
<tr>
<td>FUW</td>
<td>real</td>
<td>zonal component of advective coefficient for west face of U,V box in CLINIC and of T box in TRACER</td>
</tr>
<tr>
<td>FVN</td>
<td>real</td>
<td>meridional component of advective coefficient for north face of U,V box in CLINIC and of T box in TRACER</td>
</tr>
<tr>
<td>FVST</td>
<td>real</td>
<td>advective coefficient for south face of T box</td>
</tr>
<tr>
<td>FVSU</td>
<td>real</td>
<td>advective coefficient for south face of U,V box</td>
</tr>
<tr>
<td>FX</td>
<td>real</td>
<td>temporary value, constant in subsequent DO loops</td>
</tr>
<tr>
<td>FXA</td>
<td>real</td>
<td>temporary value, constant in subsequent DO loops</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>FXB</td>
<td>real</td>
<td>temporary value, constant in subsequent DO loops</td>
</tr>
<tr>
<td>FXC</td>
<td>real</td>
<td>temporary value, constant in subsequent DO loops</td>
</tr>
<tr>
<td>FW</td>
<td>real</td>
<td>not used; deep oceanic heat fluxes read in from the ice model</td>
</tr>
<tr>
<td>GGUJ</td>
<td>real</td>
<td>coefficient used in horizontal mixing of U,V</td>
</tr>
<tr>
<td>GICE</td>
<td>real</td>
<td>ice growth rates computed by the ice model</td>
</tr>
<tr>
<td>GM</td>
<td>real</td>
<td>masking array for U,V points; 0 for land and 1 for ocean; used in equations to force 0 values for land grid cells</td>
</tr>
<tr>
<td>GRAV</td>
<td>real</td>
<td>acceleration due to gravity = 980.6 cm/s²</td>
</tr>
<tr>
<td>GRID</td>
<td>real</td>
<td>0.5715; defined in PARAMETER statement</td>
</tr>
<tr>
<td>GRID2</td>
<td>real</td>
<td>0.28575; defined in PARAMETER statement</td>
</tr>
<tr>
<td>GRID3</td>
<td>real</td>
<td>0.28575; defined in PARAMETER statement</td>
</tr>
<tr>
<td>HHUJ</td>
<td>real</td>
<td>coefficient used in horizontal mixing of U,V</td>
</tr>
<tr>
<td>HICE</td>
<td>real</td>
<td>ice thickness computed from the ice model</td>
</tr>
<tr>
<td>HR</td>
<td>integer</td>
<td>reciprocal of total depth at U,V points</td>
</tr>
<tr>
<td>I</td>
<td>integer</td>
<td>zonal gridpoint index</td>
</tr>
<tr>
<td>IBK</td>
<td>integer</td>
<td>DO loop index</td>
</tr>
<tr>
<td>IBOX</td>
<td>integer</td>
<td>NAMELIST data file of island box corner point indices</td>
</tr>
<tr>
<td>IE</td>
<td>integer</td>
<td>ending index for I DO loop</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>IEIS</td>
<td>integer</td>
<td>ending index for island box</td>
</tr>
<tr>
<td>IEPT</td>
<td>integer</td>
<td>index designator</td>
</tr>
<tr>
<td>IEPU</td>
<td>integer</td>
<td>index designator</td>
</tr>
<tr>
<td>IEZ</td>
<td>integer</td>
<td>array of ending indices for vorticity</td>
</tr>
<tr>
<td>IFKMP</td>
<td>integer</td>
<td>number of vertical levels of ocean at T points read from bathymetry file if starting a run from scratch</td>
</tr>
<tr>
<td>ILA</td>
<td>integer</td>
<td>DO loop index</td>
</tr>
<tr>
<td>ILO</td>
<td>integer</td>
<td>DO loop index</td>
</tr>
<tr>
<td>IMT</td>
<td>integer</td>
<td>total number of T grid boxes zonally; = 360; defined in PARAMETER statement</td>
</tr>
<tr>
<td>IMTKM</td>
<td>integer</td>
<td>( MT \times KM ); = 1500; defined in PARAMETER statement</td>
</tr>
<tr>
<td>IMTM1</td>
<td>integer</td>
<td>( MT - 1 ); = 359; defined in PARAMETER statement</td>
</tr>
<tr>
<td>IMTM2</td>
<td>integer</td>
<td>( MT - 2 ); = 358; defined in PARAMETER statement</td>
</tr>
<tr>
<td>IMTP1</td>
<td>integer</td>
<td>( MT + 1 ); = 361; defined in PARAMETER statement</td>
</tr>
<tr>
<td>IMU</td>
<td>integer</td>
<td>total number of U,V grid boxes zonally; = 359; defined in PARAMETER statement</td>
</tr>
<tr>
<td>IMUM1</td>
<td>integer</td>
<td>( MU - 1 ); = 358; defined in PARAMETER statement</td>
</tr>
<tr>
<td>IMUM2</td>
<td>integer</td>
<td>( MU - 2 ); = 357; defined in PARAMETER statement</td>
</tr>
<tr>
<td>IND</td>
<td>integer</td>
<td>indicates which levels to compare for static stability</td>
</tr>
<tr>
<td>INT2</td>
<td>integer</td>
<td>1; defined in PARAMETER statement</td>
</tr>
<tr>
<td>INT3</td>
<td>integer</td>
<td>1; defined in PARAMETER statement</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>INTVL2</td>
<td>integer</td>
<td>2; defined in PARAMETER statement</td>
</tr>
<tr>
<td>INTVL3</td>
<td>integer</td>
<td>2; defined in PARAMETER statement</td>
</tr>
<tr>
<td>IPRT</td>
<td>integer</td>
<td>number of columns to print beginning at western boundary</td>
</tr>
<tr>
<td>IS</td>
<td>integer</td>
<td>starting index for I DO loop</td>
</tr>
<tr>
<td>ISIS</td>
<td>integer</td>
<td>starting I index of island box</td>
</tr>
<tr>
<td>ISLE</td>
<td>integer</td>
<td>DO loop index indicating island being computed</td>
</tr>
<tr>
<td>ISMASK</td>
<td>integer</td>
<td>gridpoint type indicator: = 0 over interior points = 1 over perimeter points = 2 over land points</td>
</tr>
<tr>
<td>ISP</td>
<td>integer</td>
<td>index designator</td>
</tr>
<tr>
<td>ISTEP</td>
<td>integer</td>
<td>number of timesteps in a day</td>
</tr>
<tr>
<td>ISTOP</td>
<td>integer</td>
<td>final column to be printed</td>
</tr>
<tr>
<td>ISTRT</td>
<td>integer</td>
<td>first column to be printed</td>
</tr>
<tr>
<td>ISZ</td>
<td>integer</td>
<td>array of starting indices for vorticity</td>
</tr>
<tr>
<td>ITT</td>
<td>integer</td>
<td>timestep counter; total number of timesteps completed</td>
</tr>
<tr>
<td>J</td>
<td>integer</td>
<td>meridional gridpoint index</td>
</tr>
<tr>
<td>JE</td>
<td>integer</td>
<td>ending index for J DO loop</td>
</tr>
<tr>
<td>JEIS</td>
<td>integer</td>
<td>ending J index of island box</td>
</tr>
<tr>
<td>JJ</td>
<td>integer</td>
<td>index designator</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>JMT</td>
<td>integer</td>
<td>total number of T grid boxes meridionally; = 360; defined in PARAMETER statement</td>
</tr>
<tr>
<td>JMTM1</td>
<td>integer</td>
<td>JMT – 1; = 359; defined in PARAMETER statement</td>
</tr>
<tr>
<td>JMTM2</td>
<td>integer</td>
<td>JMT – 2; = 358; defined in PARAMETER statement</td>
</tr>
<tr>
<td>JMTP1</td>
<td>integer</td>
<td>JMT + 1; = 361; defined in PARAMETER statement</td>
</tr>
<tr>
<td>JREV</td>
<td>integer</td>
<td>DO loop index in reverse order</td>
</tr>
<tr>
<td>JS</td>
<td>integer</td>
<td>starting index for J DO loop</td>
</tr>
<tr>
<td>JSCAN</td>
<td>integer</td>
<td>= JMTM2; defined in PARAMETER statement</td>
</tr>
<tr>
<td>JSIS</td>
<td>integer</td>
<td>starting J index of island box</td>
</tr>
<tr>
<td>K</td>
<td>integer</td>
<td>vertical gridpoint index</td>
</tr>
<tr>
<td>KAR</td>
<td>integer</td>
<td>KAR(K) = K; used to enable vectorization</td>
</tr>
<tr>
<td>KFLDS</td>
<td>integer</td>
<td>disk unit number (12) for two-dimensional horizontal fields and start and end indices</td>
</tr>
<tr>
<td>KM</td>
<td>integer</td>
<td>total number of vertical levels; = 15; defined in PARAMETER statement</td>
</tr>
<tr>
<td>KMM1</td>
<td>integer</td>
<td>KM – 1; = 14; defined in PARAMETER statement</td>
</tr>
<tr>
<td>KMP1</td>
<td>integer</td>
<td>KM + 1; = 16; defined in PARAMETER statement</td>
</tr>
<tr>
<td>KMP2</td>
<td>integer</td>
<td>KM + 2; = 17; defined in PARAMETER statement</td>
</tr>
<tr>
<td>KMT</td>
<td>integer</td>
<td>number of vertical levels of ocean at T points</td>
</tr>
<tr>
<td>KMTP</td>
<td>integer</td>
<td>KMT + 1</td>
</tr>
<tr>
<td>KMU</td>
<td>integer</td>
<td>number of vertical levels of ocean at U,V points</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>KMUP</td>
<td>integer</td>
<td>KMU + 1</td>
</tr>
<tr>
<td>KONTRL</td>
<td>integer</td>
<td>disk unit number for timestep counter, etc.</td>
</tr>
<tr>
<td>KPR</td>
<td>integer</td>
<td>temporary array for printing topography map</td>
</tr>
<tr>
<td>KREF</td>
<td>integer</td>
<td>reference level indicator</td>
</tr>
<tr>
<td>KS</td>
<td>integer</td>
<td>indicates which levels to compare for static stability</td>
</tr>
<tr>
<td>KZ</td>
<td>integer</td>
<td>temporary indicator of number of levels of ocean points</td>
</tr>
<tr>
<td>L</td>
<td>integer</td>
<td>index designator</td>
</tr>
<tr>
<td>LABS</td>
<td>integer</td>
<td>disk unit numbers (13–15) for slabs</td>
</tr>
<tr>
<td>LBC</td>
<td>integer</td>
<td>number of arrays of slab incidental data; = 2; defined in PARAMETER statement</td>
</tr>
<tr>
<td>LL</td>
<td>integer</td>
<td>index designator</td>
</tr>
<tr>
<td>LO</td>
<td>integer</td>
<td>restart tape unit number</td>
</tr>
<tr>
<td>LSEG</td>
<td>integer</td>
<td>maximum number of sets of start and end indices; = 10 for vorticity; defined in PARAMETER statement</td>
</tr>
<tr>
<td>LSEGP</td>
<td>integer</td>
<td>LSEG + 1</td>
</tr>
<tr>
<td>LU</td>
<td>integer</td>
<td>disk unit number</td>
</tr>
<tr>
<td>LUPTD</td>
<td>integer</td>
<td>permuting disk unit number for saving previous relaxation solution</td>
</tr>
<tr>
<td>LUPTDB</td>
<td>integer</td>
<td>permuting disk unit number for saving relaxation solution of two timesteps previous; if LUPTD = 5 then LUPTDB = 6; when next solution is obtained, LUPTDB is set to 5, LUPTD to 6 and the new solution is written to LUPTD</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>M</td>
<td>integer</td>
<td>index indicating the tracer being computed</td>
</tr>
<tr>
<td>MIDX</td>
<td>integer</td>
<td>100; defined in PARAMETER statement</td>
</tr>
<tr>
<td>MIDXX</td>
<td>integer</td>
<td>200; defined in PARAMETER statement</td>
</tr>
<tr>
<td>MIDDY</td>
<td>integer</td>
<td>80; defined in PARAMETER statement</td>
</tr>
<tr>
<td>MIDDYY</td>
<td>integer</td>
<td>160; defined in PARAMETER statement</td>
</tr>
<tr>
<td>MIX</td>
<td>integer</td>
<td>mixing timestep indicator; if 0, not a mixing timestep; if = 1, mixing timestep</td>
</tr>
<tr>
<td>MSB</td>
<td>integer</td>
<td>not used at present</td>
</tr>
<tr>
<td>MSCAN</td>
<td>integer</td>
<td>relaxation scan counter</td>
</tr>
<tr>
<td>MTEST</td>
<td>integer</td>
<td>print flag</td>
</tr>
<tr>
<td>MXP</td>
<td>integer</td>
<td>if = 1, second pass of Euler backward timestep</td>
</tr>
<tr>
<td>MXSCAN</td>
<td>integer</td>
<td>maximum number of relaxation scans permitted</td>
</tr>
<tr>
<td>N</td>
<td>integer</td>
<td>index designator</td>
</tr>
<tr>
<td>NA</td>
<td>integer</td>
<td>if = 1, write a restart file</td>
</tr>
<tr>
<td>NB</td>
<td>integer</td>
<td>read in by NAMELIST CONTRL</td>
</tr>
<tr>
<td>NBLK</td>
<td>integer</td>
<td>number of words per block on disk unit; argument in CSU ODAM</td>
</tr>
<tr>
<td>NBUF</td>
<td>integer</td>
<td>number of buffers set aside for disk I/O use; only used when not running in core-contained mode</td>
</tr>
<tr>
<td>NC</td>
<td>integer</td>
<td>read in by NAMELIST CONTRL</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>NCON</td>
<td>integer</td>
<td>number of passes to make on convection; since mixing is done only two levels at a time, total homogenization of unstable levels does not occur in one pass</td>
</tr>
<tr>
<td>NDICES</td>
<td>integer</td>
<td>total number of start and end indices; = 7216; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NDISK</td>
<td>integer</td>
<td>permutes with NDISKB and NDISKA on 13, 14, 15 indicating the disk units for the slabs at various time levels</td>
</tr>
<tr>
<td>NDISKA</td>
<td>integer</td>
<td>see NDISK</td>
</tr>
<tr>
<td>NDISKB</td>
<td>integer</td>
<td>see NDISK</td>
</tr>
<tr>
<td>NDISKX</td>
<td>integer</td>
<td>temporary disk unit to which data is written on second pass of Euler backward timestep</td>
</tr>
<tr>
<td>NDW</td>
<td>integer</td>
<td>unused integer read in by NAMELIST CONTDL; may be used to indicate writeout of data for analysis purposes</td>
</tr>
<tr>
<td>NE</td>
<td>integer</td>
<td>ending DO loop limit</td>
</tr>
<tr>
<td>NERGY</td>
<td>integer</td>
<td>if = 1, indicates energy printout timestep; if = 0, not an energy printout timestep</td>
</tr>
<tr>
<td>NFIRST</td>
<td>integer</td>
<td>if = 1, start a run from scratch if = 0, restart from data supplied on units 11–15; read in by NAMELIST CONTDL</td>
</tr>
<tr>
<td>NFRST</td>
<td>integer</td>
<td>disk unit address of the first word to be transferred</td>
</tr>
<tr>
<td>NGRDP</td>
<td>integer</td>
<td>= 181; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NGRDPX</td>
<td>integer</td>
<td>= 361; defined in PARAMETER statement</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>NGRDPY</td>
<td>integer</td>
<td>= 361; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NIEVEN</td>
<td>integer</td>
<td>4; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NISLE</td>
<td>integer</td>
<td>number of islands in the model basin; = 4; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NKFLDS</td>
<td>integer</td>
<td>number of two-dimensional fields needed on disk unit 12; this is six plus the number of (equivalent) two-dimensional fields needed to contain the start and end indices, normally one; = 7; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NLAST</td>
<td>integer</td>
<td>final timestep to compute on this run of the model; read in by NAMELIST CONTRL</td>
</tr>
<tr>
<td>NMIX</td>
<td>integer</td>
<td>number of timesteps between mixing timesteps; mixing is done to suppress the computational mode associated with leap-frog time-stepping</td>
</tr>
<tr>
<td>NENERGY</td>
<td>integer</td>
<td>number of timesteps between execution of energy/printout code; read in by NAMELIST CONTRL</td>
</tr>
<tr>
<td>NRTLFL1</td>
<td>integer</td>
<td>178; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NS</td>
<td>integer</td>
<td>starting DO loop limit</td>
</tr>
<tr>
<td>NSKP</td>
<td>integer</td>
<td>index designator</td>
</tr>
<tr>
<td>NSLAB</td>
<td>integer</td>
<td>number of words in one slab; = 22320; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NSTLF1</td>
<td>integer</td>
<td>0; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NSTLW1</td>
<td>integer</td>
<td>0; defined in PARAMETER statement</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>NSTR1</td>
<td>integer</td>
<td>180; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NSTUP1</td>
<td>integer</td>
<td>180; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NSWICH</td>
<td>integer</td>
<td>number of words in slab incidental data that are nonprognostic and must be switched; = 720; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NT</td>
<td>integer</td>
<td>number of tracer type variables carried in the model; temperature and salinity (2) + number of passive tracers (0); = 2; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NMIN2</td>
<td>integer</td>
<td>maximum of NT or 2; = 2; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NTOT</td>
<td>integer</td>
<td>total length of a disk unit; argument in CSU ODAM</td>
</tr>
<tr>
<td>NTNSI</td>
<td>integer</td>
<td>number of timesteps between print of a single line of information containing timestep number, kinetic energy, etc.</td>
</tr>
<tr>
<td>NULW1</td>
<td>integer</td>
<td>178; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NWDS</td>
<td>integer</td>
<td>/MT*JMT; in ODAM: number of words to transfer in a disk operation = 129600; defined in PARAMETER statement</td>
</tr>
<tr>
<td>NWRITE</td>
<td>integer</td>
<td>used in core-contained mode as number of timesteps between backup restart write</td>
</tr>
<tr>
<td>NWRS</td>
<td>real</td>
<td>number of words to transfer in a disk operation</td>
</tr>
<tr>
<td>OMEGA</td>
<td>real</td>
<td>rate of rotation of the coordinate system</td>
</tr>
<tr>
<td>P</td>
<td>real</td>
<td>mass transport stream function</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>PAD</td>
<td>namelist</td>
<td>unused memory space to prevent overwriting on I/O</td>
</tr>
<tr>
<td>PARMS</td>
<td>real</td>
<td>NAMELIST data file of various parameters in the model</td>
</tr>
<tr>
<td>PB</td>
<td>real</td>
<td>P in previous timestep</td>
</tr>
<tr>
<td>PHI</td>
<td>real</td>
<td>latitude in radians of the U,V points</td>
</tr>
<tr>
<td>PHIT</td>
<td>real</td>
<td>latitude in radians of the T points</td>
</tr>
<tr>
<td>PI</td>
<td>real</td>
<td>pi; = 3.1415927</td>
</tr>
<tr>
<td>PLICEX</td>
<td>real</td>
<td>energy change due to implicit effects on external mode</td>
</tr>
<tr>
<td>PLICIN</td>
<td>real</td>
<td>energy change due to implicit effects on internal mode</td>
</tr>
<tr>
<td>PTD</td>
<td>real</td>
<td>change of stream function across a timestep</td>
</tr>
<tr>
<td>PTDB</td>
<td>real</td>
<td>change of stream function across previous timestep</td>
</tr>
<tr>
<td>RADIUS</td>
<td>real</td>
<td>radian to degree conversion factor; = 57.29578</td>
</tr>
<tr>
<td>RADIUS</td>
<td>real</td>
<td>radius of the Earth in centimeters; 6730.65</td>
</tr>
<tr>
<td>RES</td>
<td>real</td>
<td>residual of relaxation</td>
</tr>
<tr>
<td>RESIS</td>
<td>real</td>
<td>residual of relaxation of island</td>
</tr>
<tr>
<td>RESMAX</td>
<td>real</td>
<td>maximum of the residuals at all gridpoints</td>
</tr>
<tr>
<td>RHO</td>
<td>real</td>
<td>density with a space and time invariant constant subtracted</td>
</tr>
<tr>
<td>RHON</td>
<td>real</td>
<td>RHO for the row to the north</td>
</tr>
<tr>
<td>RHOS</td>
<td>real</td>
<td>RHO for the row to the south</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RICE0</td>
<td>real</td>
<td>ratio of the latent heat of fusion of sea ice to the heat capacity of water</td>
</tr>
<tr>
<td>SCL</td>
<td>real</td>
<td>scaling factor for printout</td>
</tr>
<tr>
<td>SFU</td>
<td>real</td>
<td>external mode component of U; also used for temporary storage</td>
</tr>
<tr>
<td>SFUB</td>
<td>real</td>
<td>SFU in previous timestep</td>
</tr>
<tr>
<td>SFV</td>
<td>real</td>
<td>external mode component of V; also used for temporary storage</td>
</tr>
<tr>
<td>SFVB</td>
<td>real</td>
<td>SFV in previous timestep</td>
</tr>
<tr>
<td>SHICE</td>
<td>real</td>
<td>growth rate of ice thickness</td>
</tr>
<tr>
<td>SINE</td>
<td>real</td>
<td>sine of U,V point latitude</td>
</tr>
<tr>
<td>SINEA</td>
<td>real</td>
<td>sine of U,V point latitude in the Earth-oriented coordinates used to compute the Coriolis force</td>
</tr>
<tr>
<td>SMIX</td>
<td>real</td>
<td>interpolated monthly salinity data from the Levitus climatology</td>
</tr>
<tr>
<td>SO</td>
<td>real</td>
<td>normalizing salinities for Knudsen coefficients</td>
</tr>
<tr>
<td>SOIQ</td>
<td>real</td>
<td>normalizing salinities with alternating reference level</td>
</tr>
<tr>
<td>SOQ</td>
<td>real</td>
<td>same as SO but for vectorization</td>
</tr>
<tr>
<td>SOR</td>
<td>real</td>
<td>coefficient of over-relaxation; normally between 1.5–1.8</td>
</tr>
<tr>
<td>SORF</td>
<td>real</td>
<td>= SOR; read in NAMELIST PARMS</td>
</tr>
<tr>
<td>SQ</td>
<td>real</td>
<td>normalized salinities</td>
</tr>
<tr>
<td>STLFI</td>
<td>real</td>
<td>0; defined in PARAMETER statement</td>
</tr>
<tr>
<td>STLW1</td>
<td>real</td>
<td>0; defined in PARAMETER statement</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>STRT1</td>
<td>real</td>
<td>180; defined in PARAMETER statement</td>
</tr>
<tr>
<td>STUP1</td>
<td>real</td>
<td>180; defined in PARAMETER statement</td>
</tr>
<tr>
<td>SUMDY</td>
<td>real</td>
<td>summation of DYT</td>
</tr>
<tr>
<td>SWLDEG</td>
<td>real</td>
<td>$= -45.72$; latitude in degrees of the southern wall</td>
</tr>
<tr>
<td>SX</td>
<td>real</td>
<td>salinities; dummy argument in CSU STATE</td>
</tr>
<tr>
<td>T</td>
<td>real</td>
<td>tracer type of variables (temperature, salinity)</td>
</tr>
<tr>
<td>TA</td>
<td>real</td>
<td>similar to T but after present</td>
</tr>
<tr>
<td>TB</td>
<td>real</td>
<td>similar to T but before present</td>
</tr>
<tr>
<td>TBM</td>
<td>real</td>
<td>similar to TB but in row ( J - 1 )</td>
</tr>
<tr>
<td>TBP</td>
<td>real</td>
<td>similar to TB but in row ( J + 1 )</td>
</tr>
<tr>
<td>TBRN</td>
<td>real</td>
<td>zonal summation of tracer defined to the north</td>
</tr>
<tr>
<td>TBRS</td>
<td>real</td>
<td>zonal summation of tracer defined to the south</td>
</tr>
<tr>
<td>TBRZ</td>
<td>real</td>
<td>zonal/vertical average of tracer</td>
</tr>
<tr>
<td>TBSLAB</td>
<td>real</td>
<td>array of entire ( N-1 ) slab; equivalenced to TB</td>
</tr>
<tr>
<td>TCHENG</td>
<td>real</td>
<td>interpolated monthly Levitus data of temperature and salinity</td>
</tr>
<tr>
<td>TDIF</td>
<td>real</td>
<td>diffusion computation array</td>
</tr>
<tr>
<td>TEMPA</td>
<td>real</td>
<td>utility array used as temporary storage</td>
</tr>
<tr>
<td>TEMPB</td>
<td>real</td>
<td>utility array used as temporary storage</td>
</tr>
<tr>
<td>TEST1</td>
<td>real</td>
<td>product of the reciprocals of four surrounding depths</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>TEST2</td>
<td>real</td>
<td>sum of the reciprocals of four surrounding depths</td>
</tr>
<tr>
<td>TICE</td>
<td>real</td>
<td>ice surface temperature computed by the ice model</td>
</tr>
<tr>
<td>TINIT</td>
<td>real</td>
<td>initial values of tracers for starting from scratch</td>
</tr>
<tr>
<td>TINITF</td>
<td>real</td>
<td>= TINIT; read from NAMELIST TSPROF</td>
</tr>
<tr>
<td>TM</td>
<td>real</td>
<td>similar to T but in row $J - 1$</td>
</tr>
<tr>
<td>TMIX</td>
<td>real</td>
<td>interpolated monthly temperature data from the Levitus climatology</td>
</tr>
<tr>
<td>TMPCHG</td>
<td>real</td>
<td>temporary storage</td>
</tr>
<tr>
<td>TMT</td>
<td>real</td>
<td>meridional mass transport</td>
</tr>
<tr>
<td>TNG</td>
<td>real</td>
<td>tangent of U,V point latitude</td>
</tr>
<tr>
<td>TO</td>
<td>real</td>
<td>normalizing temperature for Knudsen coefficients</td>
</tr>
<tr>
<td>TOIQ</td>
<td>real</td>
<td>normalizing temperature with alternating reference levels</td>
</tr>
<tr>
<td>TOQ</td>
<td>real</td>
<td>similar to TO but for vectorization</td>
</tr>
<tr>
<td>TOTDX</td>
<td>real</td>
<td>total zonal span of ocean boxes</td>
</tr>
<tr>
<td>TOTDZ</td>
<td>real</td>
<td>total vertical span of ocean boxes</td>
</tr>
<tr>
<td>TP</td>
<td>real</td>
<td>similar to T but in row $J + 1$</td>
</tr>
<tr>
<td>TQ</td>
<td>real</td>
<td>normalized temperature</td>
</tr>
<tr>
<td>TSLAB</td>
<td>real</td>
<td>array of entire $N$ slab; equivalenced to T in CSU</td>
</tr>
<tr>
<td>TSPROF</td>
<td>namelist</td>
<td>NAMELIST data file for initial tracers</td>
</tr>
<tr>
<td>TSTEPS</td>
<td>namelist</td>
<td>NAMELIST data file for timestep lengths</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>TTDAY</td>
<td>real</td>
<td>current day of the year number</td>
</tr>
<tr>
<td>TTDTOT</td>
<td>real</td>
<td>array of integrals on tracers</td>
</tr>
<tr>
<td>TTN</td>
<td>real</td>
<td>northward transport of tracers</td>
</tr>
<tr>
<td>TTSEC</td>
<td>real</td>
<td>current total elapsed time in seconds</td>
</tr>
<tr>
<td>TTYEAR</td>
<td>real</td>
<td>number of years of integration completed</td>
</tr>
<tr>
<td>TVAR</td>
<td>real</td>
<td>change of variance of tracers</td>
</tr>
<tr>
<td>TX</td>
<td>real</td>
<td>temperature in CSU STATE; temporary array in CSU STRESSUP</td>
</tr>
<tr>
<td>U</td>
<td>real</td>
<td>zonal component of velocity</td>
</tr>
<tr>
<td>UA</td>
<td>real</td>
<td>similar to U but for timestep after present</td>
</tr>
<tr>
<td>UB</td>
<td>real</td>
<td>similar to U but for timestep before present</td>
</tr>
<tr>
<td>UBM</td>
<td>real</td>
<td>similar to UB but in row J - 1</td>
</tr>
<tr>
<td>UBP</td>
<td>real</td>
<td>similar to UB but in row J + 1</td>
</tr>
<tr>
<td>UCLIN</td>
<td>real</td>
<td>array of internal mode component of U at row J + 1</td>
</tr>
<tr>
<td>UDIF</td>
<td>real</td>
<td>vertical mixing computation array for U</td>
</tr>
<tr>
<td>UENG</td>
<td>real</td>
<td>individual forcing terms on U</td>
</tr>
<tr>
<td>UICE</td>
<td>real</td>
<td>ice-drift velocity of three timesteps computed by the ice model</td>
</tr>
<tr>
<td>UICEC</td>
<td>real</td>
<td>immediate ice-drift velocity computed by the ice model</td>
</tr>
<tr>
<td>UM</td>
<td>real</td>
<td>similar to U but in row J - 1</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>UOVER</td>
<td>real</td>
<td>positioned in common to be equivalent of UDIF with $K$ index equal to $0$; used to set surface boundary conditions</td>
</tr>
<tr>
<td>UP</td>
<td>real</td>
<td>U in row $J + 1$</td>
</tr>
<tr>
<td>USAV</td>
<td>real</td>
<td>array of internal mode component of $U$ at row $J$</td>
</tr>
<tr>
<td>UUNDER</td>
<td>real</td>
<td>positioned in common to be equivalent of UDIF with $K$ index equal to $KM + 1$</td>
</tr>
<tr>
<td>V</td>
<td>real</td>
<td>meridional component of velocity</td>
</tr>
<tr>
<td>VA</td>
<td>real</td>
<td>$V$ at timestep after present</td>
</tr>
<tr>
<td>VB</td>
<td>real</td>
<td>$V$ at timestep before present</td>
</tr>
<tr>
<td>VBM</td>
<td>real</td>
<td>similar to $VB$ but in row $J - 1$</td>
</tr>
<tr>
<td>VBP</td>
<td>real</td>
<td>similar to $VB$ but in row $J + 1$</td>
</tr>
<tr>
<td>VBR</td>
<td>real</td>
<td>zonal average of horizontal velocity</td>
</tr>
<tr>
<td>VBRZ</td>
<td>real</td>
<td>zonal/vertical average of horizontal velocity</td>
</tr>
<tr>
<td>VCLIN</td>
<td>real</td>
<td>array of internal mode component of $V$ at row $J + 1$</td>
</tr>
<tr>
<td>VDIFF</td>
<td>real</td>
<td>vertical mixing computation array for $V$</td>
</tr>
<tr>
<td>VENG</td>
<td>real</td>
<td>individual forcing terms on $V$</td>
</tr>
<tr>
<td>VICE</td>
<td>real</td>
<td>ice-drift velocity of three timesteps computed by the ice model</td>
</tr>
<tr>
<td>VICEC</td>
<td>real</td>
<td>immediate ice-drift velocity computed by the ice model</td>
</tr>
<tr>
<td>VM</td>
<td>real</td>
<td>$V$ in row $J - 1$</td>
</tr>
<tr>
<td>VOLUME</td>
<td>real</td>
<td>total volume of the model basin</td>
</tr>
<tr>
<td>DATA ELEMENT</td>
<td>DATA TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>VOVER</td>
<td>real</td>
<td>positioned in common to be equivalent of VDIF with $K$ index equal to 0; used to set surface boundary conditions</td>
</tr>
<tr>
<td>VP</td>
<td>real</td>
<td>$V$ in row $J+1$</td>
</tr>
<tr>
<td>VSAV</td>
<td>real</td>
<td>array of internal mode component of $V$ at row $J$</td>
</tr>
<tr>
<td>VUNDER</td>
<td>real</td>
<td>positioned in common to be equivalent of VDIF with $K$ index equal to $KM+1$</td>
</tr>
<tr>
<td>W</td>
<td>real</td>
<td>vertical velocity; computed in vertical line with $U,V$ in CLINIC; computed in vertical line with $T$ in TRACER</td>
</tr>
<tr>
<td>WINDSX</td>
<td>real</td>
<td>zonal wind velocity</td>
</tr>
<tr>
<td>WINDSY</td>
<td>real</td>
<td>meridional wind velocity</td>
</tr>
<tr>
<td>WSX</td>
<td>real</td>
<td>zonal component of surface wind stress</td>
</tr>
<tr>
<td>WSXM</td>
<td>real</td>
<td>WSX in row $J-1$</td>
</tr>
<tr>
<td>WSXP</td>
<td>real</td>
<td>WSX in row $J+1$</td>
</tr>
<tr>
<td>WSY</td>
<td>real</td>
<td>meridional component of surface wind stress</td>
</tr>
<tr>
<td>WSYM</td>
<td>real</td>
<td>WSY in row $J-1$</td>
</tr>
<tr>
<td>WSYP</td>
<td>real</td>
<td>WSY in row $J+1$</td>
</tr>
<tr>
<td>YNEG</td>
<td>real</td>
<td>temporary storage used to read negative ice to be melted; heat above the freezing temperature computed by the ice model</td>
</tr>
<tr>
<td>ZDZ</td>
<td>real</td>
<td>vertical position of bottom of levels</td>
</tr>
<tr>
<td>ZDZZ</td>
<td>real</td>
<td>vertical position of center of levels</td>
</tr>
<tr>
<td>ZTD</td>
<td>real</td>
<td>change of vorticity across one timestep</td>
</tr>
</tbody>
</table>
### DATA ELEMENT | DATA TYPE | DESCRIPTION | CSU DEFINED IN | CSU USED IN | COMMON BLOCK
--- | --- | --- | --- | --- | ---
ZUN | real | time change of vertically averaged zonal forcing at north face | CLINIC | CLINIC | ONEDIM
ZUNENG | real | vertical average of U forcing at north face | CLINIC | CLINIC | WORKSP
ZUS | real | time change of vertically averaged zonal forcing at south face | CLINIC | CLINIC | ONEDIM
ZUSENG | real | vertical average of V forcing at south face | CLINIC | CLINIC | WORKSP
ZVN | real | time change of vertically averaged meridional forcing at north face | CLINIC | CLINIC | ONEDIM
ZVENG | real | vertical average of V forcing at north face | CLINIC | CLINIC | WORKSP
ZVS | real | time change of vertically averaged meridional forcing at south face | CLINIC | CLINIC | ONEDIM
ZVSENG | real | vertical average of V forcing at south face | CLINIC | CLINIC | WORKSP

#### 5.0 CSCI PIPS2.0 DATA FILES

This section describes the transfer of data elements between PIPS2.0 CSCs and CSUs via data/disk files.

#### 5.1 Data File to CSU/CSC Cross-Reference

<table>
<thead>
<tr>
<th>DATA FILE</th>
<th>UNIT #</th>
<th>READ FROM:</th>
<th>WRITTEN TO:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Levitus Temperature</td>
<td>10</td>
<td>CSU DRIVER</td>
<td></td>
</tr>
<tr>
<td>Levitus Salinity</td>
<td>11</td>
<td>CSU DRIVER</td>
<td></td>
</tr>
<tr>
<td>NOGAPS</td>
<td>12</td>
<td>CSU RFORCE</td>
<td></td>
</tr>
<tr>
<td>River Discharge</td>
<td>19</td>
<td>CSU DRIVER</td>
<td></td>
</tr>
<tr>
<td>Grid Positions</td>
<td>14</td>
<td>CSU DRIVER</td>
<td></td>
</tr>
<tr>
<td>Land/Sea Masks</td>
<td>15</td>
<td>CSU BNDRY</td>
<td>CSU RST_OCN</td>
</tr>
<tr>
<td>Ocean Data</td>
<td>13</td>
<td>CSU RST_OCN</td>
<td>CSU SAV_OCN</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ice Restart Data</td>
<td>16</td>
<td>CSU RST ICE</td>
<td>CSU SAV ICE</td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ocean Restart Data</td>
<td>18</td>
<td>CSU ODAM</td>
<td>CSU ODAM</td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td>CSU DRIVER</td>
</tr>
<tr>
<td>Graphics Data</td>
<td>31</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.2 CSU ODAM Virtual Disk Access Logical Units

5.2.1 Logical Unit Number KONTRL

**Purpose:** Logical unit number KONTRL shall contain the timestep counter, total elapsed time and area and volume of the basin.

**Size:** The maximum size of the file will be 20 words.

**File Access Method:** Direct Access (simulated by CSU ODAM).

**Data Stored in File (array BIG):**

- **ITT** (timestep counter; total number of timesteps completed)
- **TTSEC** (total elapsed time in seconds)
- **AREA** (area of the ocean basin)
- **VOLUME** (volume of the ocean basin)
- **AKNTRL(6)** (available for storage of values that need to be saved to the restart file)
- **PAD(10)** (used to prevent overwriting on I/O in core-contained mode)

5.2.2 Logical Unit Number KFLDS

**Purpose:** Logical unit number KFLDS shall contain CSCI Ocean two-dimensional horizontal fields.

**Size:** The maximum size of the file will be 907,200 words.

**File Access Method:** Direct Access (simulated by CSU ODAM).

**Data Stored in File (array BIG):**

- **P** (mass transport stream function)
- **PB** (mass transport stream function for the previous timestep)
- **ZTD** (change of vorticity across one timestep)
- **HR** (reciprocals of total depth at U,V points)

5.2.3 Logical Unit Numbers LABS(13), LABS(14), LABS(15)

**Purpose:** Logical unit numbers LABS(13), LABS(14), and LABS(15) contains the primary slab data for the \( N - 1, N, \) and \( N + 1 \) timesteps, respectively.

**Size:** The maximum size of each file will be 22,320 words.

**File Access Method:** Direct Access (simulated by CSU ODAM).

**Data Stored in File (array BIG):**

Both primary slab data and slab incidental data are stored in these files. The primary slab data includes tracer data and U and V components of horizontal velocity. The slab incidental data includes the number of vertical levels of the ocean at T and U,V points and the wind stress data. Primary slab data is three-dimensional with meridional, zonal, and vertical dimensions. Base arrays T, U, and V contain primary slab data. Slab incidental data is two-dimensional with meridional and zonal dimensions. FKMTP, FKMUP, WSX, and WSY are arrays of slab incidental data. Because of the
Table 5.2.3-1 — Six-Step Cycle Summarizes the Permuting Disk Units Execute

<table>
<thead>
<tr>
<th>UNIT NUMBER</th>
<th>Timestep 1</th>
<th>Timestep 2</th>
<th>Timestep 3</th>
<th>Timestep 4</th>
<th>Timestep 5</th>
<th>Timestep 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>$N-1$, A</td>
<td>$N+1$, B</td>
<td>$N$, B</td>
<td>$N-1$, B</td>
<td>$N+1$, A</td>
<td>$N$, A</td>
</tr>
<tr>
<td>14</td>
<td>$N$, B</td>
<td>$N-1$, B</td>
<td>$N+1$, A</td>
<td>$N$, A</td>
<td>$N-1$, A</td>
<td>$N+1$, B</td>
</tr>
<tr>
<td>15</td>
<td>$N+1$, A</td>
<td>$N$, A</td>
<td>$N-1$, A</td>
<td>$N+1$, B</td>
<td>$N$, B</td>
<td>$N-1$, B</td>
</tr>
</tbody>
</table>

Permuting disk I/O used to handle the data, the meridional or row dimension of the arrays is limited to three.

On timestep 1 and every sixth timestep, logical units 13, 14, and 15 contain the following data:

**Logical Unit 13:**
- **TBP**: tracer data for $N-1$ timestep for row $J+1$
- **UBP**: zonal component of horizontal velocity for $N-1$ timestep for row $J+1$
- **VBP**: meridional component of horizontal velocity for $N-1$ timestep for row $J+1$
- **FKMUP**: number of vertical levels of ocean at U,V points for row $J+1$
- **WSYP**: meridional component of surface wind stress for row $J+1$

**Logical Unit 14:**
- **TP**: tracer data for $N$ timestep for row $J+1$
- **UP**: zonal component of horizontal velocity for $N$ timestep for row $J+1$
- **VP**: meridional component of horizontal velocity for $N$ timestep for row $J+1$
- **FKMTP**: number of vertical levels of ocean at U,V points for row $J+1$
- **WSXP**: zonal component of surface wind stress for row $J+1$

**Logical Unit 15:**
- **TA**: tracer data for $N+1$ timestep
- **UA**: zonal component of horizontal velocity for $N+1$ timestep
- **VA**: meridional component of horizontal velocity for $N+1$ timestep
- **BCON**: used to write out FKMU and WSY on odd timestep, FKMTP and WSX on even timestep

Table 5.2.3-1 summarizes the 6-step cycle that the permuting disk units execute. In this table, $N-1$, $N$, and $N+1$ denote primary slab data for the $N-1$ (timestep before present), $N$ (present), and $N+1$ (timestep after present) timesteps. Slab incidental data is denoted by A and B. FKMU and WSY are A type arrays and FKMTP and WSX are B type arrays. On even timesteps, the A and B type data are read into incorrect arrays and must be switched.

## 6.0 ABBREVIATIONS AND ACRONYMS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI</td>
<td>Computer Software Configuration Item</td>
</tr>
<tr>
<td>CSC</td>
<td>Computer Software Component</td>
</tr>
<tr>
<td>CSU</td>
<td>Computer Software Unit</td>
</tr>
<tr>
<td>NOGAPS</td>
<td>Navy Operational Global Atmospheric Prediction System</td>
</tr>
<tr>
<td>NRL</td>
<td>Naval Research Laboratory</td>
</tr>
</tbody>
</table>
7.0 SUMMARY AND CONCLUSION

Since 1987, FNMOC has been running sea-ice forecasting systems in various regions of Navy interest (Central Arctic, the Barents Sea, and the Greenland Sea). The Polar Ice Prediction System (PIPS1.1) predicts sea ice conditions in the Arctic basin, the Barents Sea, and the Greenland Sea at a resolution of 127 km. Two regional sea-ice forecasting systems, the Polar Ice Prediction System—Barents (RPIPS-B) and the Polar Ice Prediction System—Greenland Sea (RPIPS-G), also predict sea ice conditions in the Barents Sea and the Greenland Sea, respectively, at a higher resolution of 20–25 km. In 1995, NRL delivered to FNMOC a coupled ice-ocean system, PIPS2.0, which predicts sea ice conditions for most of the ice-covered regions in the Northern Hemisphere. PIPS2.0 will replace the existing three operational forecast systems when it completes the final operational testing phase at FNMOC. PIPS2.0 uses as its basis the Hibler ice model and the COG ocean model. PIPS2.0 has a resolution of approximately a quarter of a degree, which is similar to the resolution of the operational regional systems (RPIPS-B and RPIPS-G).

8.0 ACKNOWLEDGMENTS

This work was funded by the U.S. Space and Naval Warfare Systems Command, Program Element 0603207N.

9.0 REFERENCES


