Technical Report No. 13

The Hamburg Ocean Primitive Equation Model

H O P E

Jörg-Olaf Wolff , Antarctic CRC, Hobart, Australia;
Ernst Maier-Reimer, Max-Planck-Institut für Meteorologie
Stephanie Legutke, Deutsches Klimarechenzentrum
Bundesstraße 55, D-20146 Hamburg

Edited by: Modellberatungsgruppe, DKRZ
Hamburg, April 1997
ISSN 0940-9327

Table of Contents

DKRZ HOPE Model Documentation

List of Figures

DKRZ HOPE Model Documentation

ABSTRACT

HOPE is a primitive-equation model of the global ocean circulation, but may also be used for regional studies. Prognostic variables are the three-dimensional horizontal velocity fields, the sea-surface elevation, and the thermohaline variables. The equations are discretized on prescribed horizontal surfaces using Arakawa-E-type grids. Two time levels are used for the integration.

A Hibler-type dynamic sea-ice model allows a prognostic calculation of sea-ice thickness, compactness, and velocities. The thermodynamic growth of sea ice is calculated with heat balance equations using simple bulk formulae. A snow layer is included.

HOPE is in particular useful for altimetry data assimilation purposes because of the prognostic calculation of the sea-surface elevation.

The HOPE model has been written by Ernst Maier-Reimer at the Max-Planck Institute for Meteorology in Hamburg, Germany.

Contributions to the model code / postprocessing / diagnostics are also due to

Otto Boehringer

Sybren Drijfhout

Christoph Heinze

Arno Hellbach

Mojib Latif

Stephanie Legutke

Tim Stockdale

Andreas Sterl

Achim Stoessel

Joerg-Olaf Wolff.

1 Introduction

The “Hamburg Ocean Primitive Equation” model, HOPE, is an ocean general circulation model (OGCM) based on the primitive equations with representation of thermodynamic processes. It is capable of simulating the oceanic circulation from small scales (oceanic eddies) to gyre scales, in response to atmospheric forcing fields. For an application on horizontal scales smaller than say 1 km the hydrostatic assumption is no longer valid and the model must in parts be reformulated.

The use of an ocean circulation model requires a comprehensive understanding of the ocean physics and the numerical formulation. It is not recommended to use this model as a black box. Many physical processes in the ocean are still not very well understood and are therefore only crudely parameterized. Each new application requires a new consideration of how to specify model parameters, especially coefficients for eddy viscosity and diffusivity. The model is thought to be a framework into which new ideas concerning parameterisations or forcing mechanisms might easily be incorporated.

This manual gives a description of the HOPE model, in order to help potential users to acquire a thorough understanding of the model physics and the numerics.

The manual commences with a short description of the physics and an introduction to the grid used for the spatial finite differencing. Then, a full time step is discussed in terms of the subroutines used in the course of the integration, including different solution algorithms if available. Section 9 “User’s manual” intends to provide the user with all information relevant to run the model in a geometry of his or her choice. A list of model variables is added at the end of the manual.

The HOPE subroutines together with compile-scripts, forcing and initial data, and example scripts for the integration of the model with specific model set ups are available at the DKRZ.

A postprocessing package based on GrADS is available as well.

2 Model physics

HOPE is based on the non-linear balance equations for momentum, the continuity equation for an incompressible fluid, and conservation equations for heat and salt. The hydrostatic and Boussinesq approximations are applied. Prognostic ocean variables are horizontal velocities, sea-surface elevation, potential temperature, and salinity.

The horizontal momentum balance equations are:

$$\frac{d}{dt}\vec{v} + f(\vec{k}\times\vec{v}) = -\frac{1}{\rho_o}(\vec{\nabla}_H(p+\rho_o g\zeta)) + \vec{F}_H + \vec{F}_V\,. \tag{2.1}$$

$\vec{v} = (u, v)$ is the horizontal velocity vector, f the Coriolis parameter, $\vec{k}$ an upward vertical unit vector, ρ_o a constant reference density, g the gravitational acceleration of the earth, and $p = g\int_z^0 \rho dz'$ the internal pressure. ζ is the sea-surface elevation, $\frac{d}{dt} = \frac{\partial}{\partial t} + \vec{v}\cdot\vec{\nabla}_H + w\cdot\frac{\partial}{\partial z}$ the total derivative, and $\vec{\nabla}_H$ the horizontal gradient operator.

The horizontal and vertical turbulent viscous terms $\vec{F}_H$ and $\vec{F}_V$ are specified according to K-theory. A number of formulations can be used in combination. These are harmonic and biharmonic formulations

$$\vec{F}_{H1} = A_H\nabla^2(\vec{v}) \tag{2.2}$$

$$\vec{F}_{H2} = -B_H\nabla^4(\vec{v}), \tag{2.3}$$

and a viscous dissipation depending on the local rate of strain

$$\vec{F}_{H3} = \nabla(\nu_A T^2\vec{\nabla}_H(\vec{v})). \tag{2.4}$$

A_H, B_H, and ν_A are constant coefficients, ∇ is the divergence, $\nabla^2 = \nabla\cdot\vec{\nabla}_H$ the horizontal Laplace operator, and T^2 is a function of the local rate of strain $\partial v/\partial x + \partial u/\partial y$.

The vertical eddy viscosity is parameterized as

$$\vec{F}_V = \frac{\partial}{\partial z}\left(A_V\frac{\partial}{\partial z}\vec{v}\right), \tag{2.5}$$

where the eddy coefficient A_V can be specified to depend on the local Richardson number.

The internal pressure p is related to the density field by the hydrostatic equation

$$\frac{\partial p}{\partial z} = -g\rho\,. \tag{2.6}$$

The density is calculated with a non-linear polynomial depending on salinity, temperature, and pressure (UNESCO, 1983). The UNESCO formula for density must be used with in-situ temperatures. This is accounted for by the subroutine ADISIT, which converts potential temperatures into in-situ temperatures.

The vertical velocity w is calculated diagnostically from the incompressibility condition

$$\frac{\partial w}{\partial z} = -\nabla\vec{v}\,. \tag{2.7}$$

In cases of unstable stratification, convective adjustment is applied, i.e. each pair of vertically adjacent unstably stratified water cells is mixed with heat and salt being conserved.

The sea-surface elevation is calculated from the linearized kinematic boundary condition

$$\frac{\partial \zeta}{\partial t} = w|_{z=0} + Q_\zeta = -\nabla\int_{-H}^{0}\vec{v}dz + Q_\zeta\,, \tag{2.8}$$

where $H(x, y)$ is the water depth, and Q_ζ the fresh-water forcing of the free surface.

At the sea surface, a momentum flux must be specified:

$$\rho_o A_V \frac{\partial}{\partial z}\vec{v}\Big|_{z=0} = \vec{\tau}\,. \tag{2.9}$$

In addition to no-slip conditions at lateral boundaries, a Newtonian friction law is applied. Bottom friction is also computed by assuming Newtonian friction.

Salinity S and potential temperature θ are determined from the continuity equations

$$\frac{dS}{dt} = \frac{\partial}{\partial z}\left(D_V\frac{\partial S}{\partial z}\right) + D_H\nabla^2(S) + \nabla(\nu_D T^2\vec{\nabla}_H(S))\,, \tag{2.10}$$

$$\frac{d\theta}{dt} = \frac{\partial}{\partial z}\left(D_V\frac{\partial \theta}{\partial z}\right) + D_H\nabla^2(\theta) + \nabla(\nu_D T^2\vec{\nabla}_H(\theta)) \tag{2.11}$$

where ν_D and D_H are constant coefficients, and T^2 is defined as for the horizontal viscosity. The vertical eddy diffusivity D_V can be specified to depend on the local Richardson number.

Surface boundary conditions for salinity can be a combination of the following (see section 4.1):

i) relaxation to a specified salinity field S_{obs}

$$D_V \frac{\partial S}{\partial z}\bigg|_{z=\zeta} = \lambda_S (S_{obs} - S_1) \tag{2.12}$$

ii) salinity changes according to specified freshwater fluxes

iii) If the sea-ice model is active (*ICESEA = 1)*, the surface salinity changes according to precipitation - evaporation, snow fall or melt, and ice growth or melt are calculated in subroutine GROWTH.

If condition i) is applied, a corresponding freshwater flux is calculated as well (see equation (4.1.7)). Condition i) might be combined with ii) or iii). Freshwater fluxes add to the surface elevation.

The potential temperature boundary condition at the surface can be specified

i) by relaxation to an observed surface temperature field θ_{obs}:

$$D_V \frac{\partial \theta}{\partial z}\bigg|_{z=\zeta} = \lambda_\theta (\theta_{obs} - \theta_1) \tag{2.13}$$

ii) by applying a specified heat flux

iii) If the sea-ice model is active (*ICESEA = 1)*, the heat flux into the surface layer is calculated in subroutine GROWTH.

Condition i) can be combined with ii) or iii). In the present version, condition ii) is used in ice free regions, while iii) is used in regions potentially covered by ice. This has to be chosen according to the forcing data that is used.

At the lateral boundaries and at the sea floor, zero-flux conditions are used.

3 Numerical scheme

The numerical scheme is based on two time levels and a (variable) number of vertical levels. For the horizontal discretisation, an Arakawa E-grid (Arakawa and Lamb, 1977) is used. It is divided into two subgrids, one for the even rows and one for the odd rows, as indicated in Figure 1.

3.1 Horizontal discretisation

The model variables are defined on a horizontal staggered grid (Figure 1).

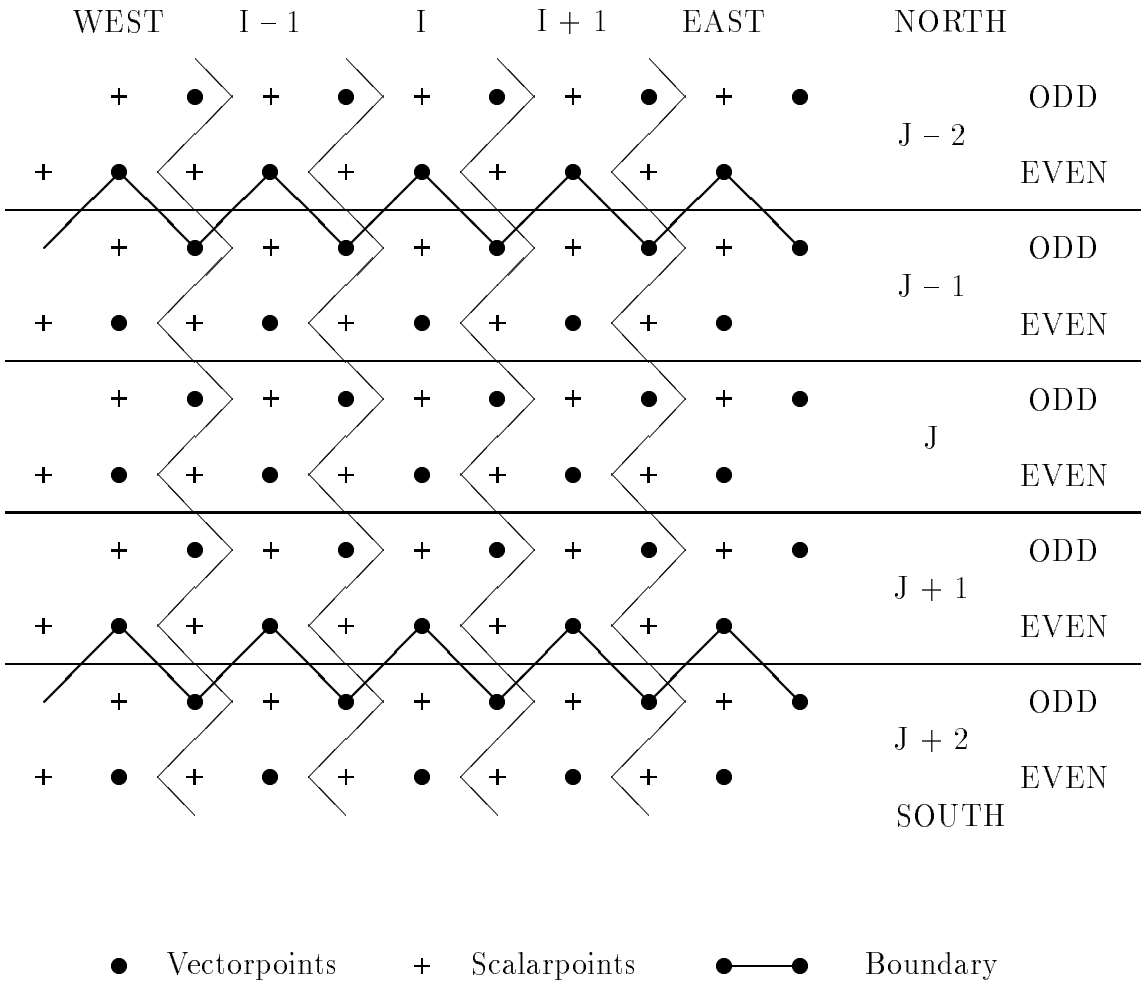


Figure 1 **Arakawa E-grid with ODD/EVEN rows**

Variables defined on vector points are:

- horizontal ocean velocities $\vec{v} = (u, v)$
- wind stress $\vec{\tau} = (\tau^x, \tau^y)$
- sea ice velocities $\vec{v_I} = (u_I, v_I)$.

Variables defined on scalar points are:

- potential temperature θ
- salinity S
- density ρ
- internal pressure p
- vertical velocity w (vertically staggered, see section 3.2)
- coefficients of vertical viscosity and diffusivity (vertically staggered, see section 3.2)
- sea-surface elevation ζ
- heat and freshwater fluxes across the air/sea interface
- sea ice thickness h_I and compactness A_I
- snow depth h_s.

Multiple grid refinements on an orthogonal grid are possible. HOPE calculates a variable grid using arrays containing information on longitudes and latitudes. The grid-point distances $\Delta x(x, y)$ and $\Delta y(x, y)$, which take into account the spherical geometry of the earth, are stored on 2-dimensional arrays (see Table 1).

		●	+
ODD	latitude	ALAT(2J-1)	ALAT(2J-1)
ODD	longitude	ALONG(2I+2)	ALONG(2I+1)
EVEN	latitude	ALAT(2J)	ALAT(2J)
EVEN	longitude	ALONG(2I+1)	ALONG(2I)

Table 1 : Storage of geographical positions of grid points

Indexing is done as follows (for each subgrid EVEN/ODD):

- West - East: $I = 1, \ldots, IE$
- North - South: $J = 1, \ldots, JE$
- Sea surface - Bottom: $K = 1, \ldots, KE$

For the case of periodic boundaries (with parameter $ICYCLI = 1$), the grid cells with $I = 1$ and $I = 2$ are identified to those with $I = IE - 1$ and $I = IE$, respectively.

3.2 Vertical discretisation

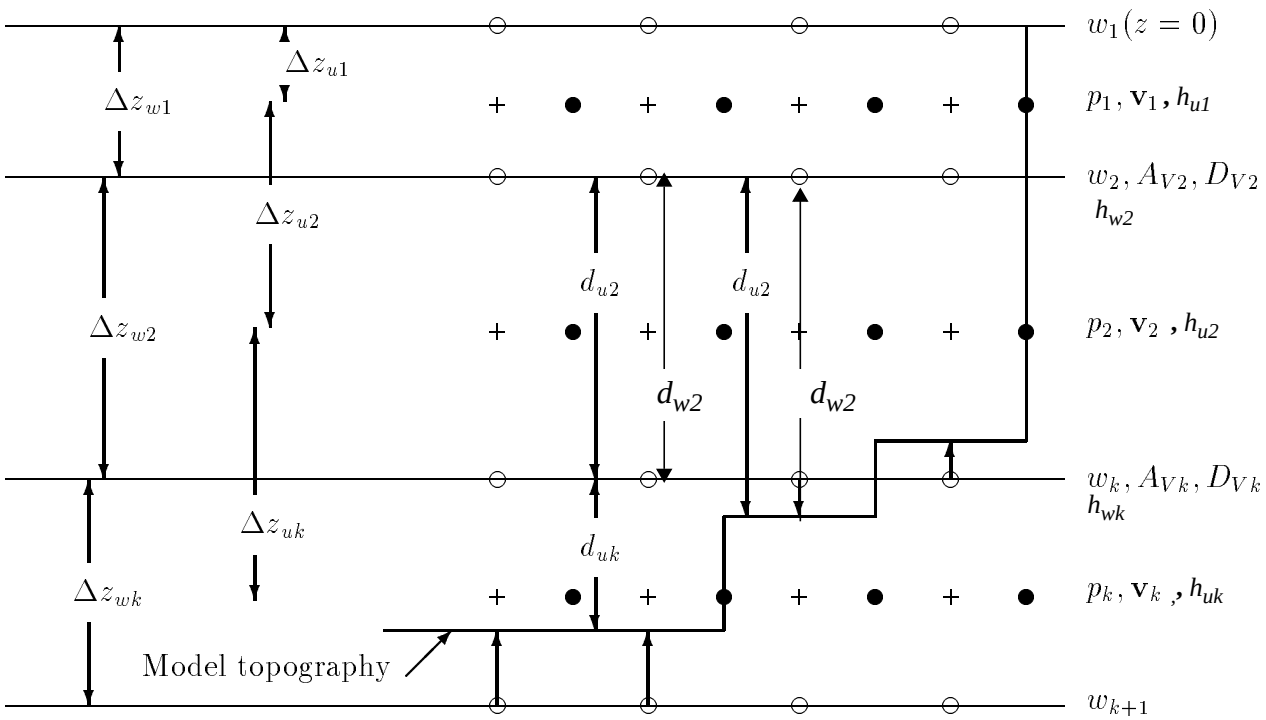


Figure 2 **Vertical structure of the model grid**

Grid-box thicknesses are the same throughout each layer except for the lowest box in each column, whose thickness is adapted to the local topography. The vertical velocities are computed at points that are shifted horizontally and vertically relative to the points, where the horizontal velocity components are computed (Figures 1 and 2).

The vertical grid is defined by [Array (*dimensions*): description]:

- $DZW(KE)$: Vertical distances Δz_{wk} between two consecutive *w*-surfaces (= standard layer thickness). The layer thicknesses are set in the main program via *DATA DZW* /. . . / and may be changed by the user.
  All other vertical distances are computed from this array as given below.

- $TIESTW(KE+1)$: Depth h_w of *w*- surfaces

$$TIESTW(1) = 0 \qquad .$$

$$TIESTW(k) = \sum_{l=2}^{k} \Delta z_{w(l-1)} \qquad k = 2, \ldots, KE \qquad (3.2.1)$$

- $TIESTU(KE+1)$: Depth h_u of $\dot{v}$, p-surfaces

$$TIESTU(k) = 0.5(TIESTW(k+1) + TIESTW(k)) \qquad k = 1,\ldots,KE \tag{3.2.2}$$

- $DZ(KE+1)$: Vertical distances Δz_{uk} between two $\dot{v}$, p-surfaces

$$\begin{aligned} DZ(1) &= \frac{1}{2}DZW(1) \\ DZ(k) &= TIESTU(k) - TIESTU(k-1) \qquad k = 2,\ldots,KE+1 \end{aligned} \tag{3.2.3}$$

After a topography dataset has been provided on the grid specified by the user (at the positions of vector points), HOPE recalculates the actual depth of scalar points as the maximum depth of the four surrounding vector points.

The individual grid box thicknesses are stored on arrays

- $DDU_{E/O}(I,J,K) = d_{uk}$, $I=1,IE$, $J=1,JE$, $K=1,KE$ for vector points
- $DDP_{E/O}(I,J,K) = d_{wk}$, $I=1,IE$, $J=1,JE$, $K=1,KE$ for scalar points,

where *E/O* indicate the EVEN/ODD rows with north-south index *J*.

$d_{w1}+\zeta-h_{draft}$ is used instead of d_{w1}, where appropriate. h_{draft} is the ice and snow draft.

3.3 Time stepping method

The time stepping in HOPE is based on the concept of operator splitting, which is also called *time splitting* or *method of fractional steps*.

The method is illustrated with the following example (taken from Press et al., 1986).
Suppose you have an initial value equation of the form

$$\frac{\partial u}{\partial t} = \Im u \tag{3.3.1}$$

where $\Im$ is an arbitrary operator, which can be non-linear. We will assume that this operator can be divided into m pieces, each acting separately on u,

$$\Im u = (\Im_1 + \Im_2 + \ldots \Im_m)u \tag{3.3.2}$$

In the numerical scheme, $\sum_{i=1}^{m} \Im_i$ is approximated by the sequential application of discrete operators

$$(\Re_m(\Re_{m-1}(\Re_{m-2}(\ldots \Re_1)))\ldots)u^n \approx \left(\sum_{i=1}^{m} \Im_i\right)u^n \text{, i.e.}$$

$$\frac{u^n - u^{n-1}}{\Delta t} = (\Re_m(\Re_{m-1}(\Re_{m-2}(\ldots \Re_1)))\ldots)u^n$$

Thus, the numerical model can be written in modular structure (i.e. successive updates of one variable in several subroutines).

In this manual, the notation of Press et al. (1986) is used, where $u^{n+k/m}$ denotes the update of u^n after the application of the operator $\Re_k$.

Diagnostic variables, e.g. *w,* are always computed with the last update available of the relevant prognostic variables.

4 Time stepping

The time stepping is controlled by the main program OCEMAST**.** First, all variables are initialized in subroutine OCEINI. The model is advanced one time step by each call of subroutine OCESTEP. Flow charts of the model system can be found in “Flow charts” on page 68.

OCESTEP calls the following subroutines (in the order listed):

Subroutine	Description
OCTHER	Thermohaline forcing (if *ICESEA = 1*, the sea-ice model is called) Baroclinic pressure Convective adjustment Vertical eddy viscosity and diffusivity coefficients
OCWIND	Wind forcing
OCVISC	Vertical momentum advection Vertical eddy viscosity Horizontal eddy viscosity Bottom friction
OCBIHAE	Called if *BHFR > 1.E-12* Horizontal eddy viscosity (biharmonic friction with rotated operator)
OCBIHAR	Called if *BHF > 1.E-12* Horizontal eddy viscosity (biharmonic friction)
OCIADJ7	Horizontal momentum advection
OCMODS	Separation into barotropic and baroclinic velocity
OCCLIT_ GEOPOL	Solution of the baroclinic system (assuming geostrophy near the pole)
OCRHSZ	Computation of the right-hand side of the barotropic system
ZGAUSS/ ZGAUSS2	Direct solution of the barotropic system for *IELIMI = 1* (ZGAUSS/ZGAUSS2 are used if the matrix is read from disk/kept in memory(*IMATR=1/0*))
ZSOR	Iterative solution of the barotropic system for $IELIMI = 0$
OCVTRO	New barotropic velocities
OCVTOT	New total velocities *u*,*v* and *w*
OCADUP	Called if $ITSADV = 0$. 3-D advection of thermohaline fields (upstream) Horizontal thermohaline diffusion

Subroutine	Description
OCADV	Called if *ITSADV = 1*. 3-D advection of thermohaline fields (predictor-corrector, central differencing) Horizontal thermohaline diffusion
OCVERDI	Vertical thermohaline diffusion
OCPODI	Called if *NPOLE = 1*. Smoothing at the North Pole

In the following, these time-stepping subroutines will be described (subscript *k* denotes the layer; symbols are explained in Appendix B).

A description of the sea-ice model which is called in subroutine OCTHER when *ICESEA=1* is found in section 8.

Auxiliary subroutines and functions are listed in section 11.
Subroutines which are available for diagnostic purposes are also listed there.

The time indexing (superscript) is as follows: *n* denotes the variables at the last time level, *n+1* denotes the variables at the end of the actual time step, and *n+l/m* denotes the update of the variable by the *l*-th operator (see section 3.3).

4.1 Subroutine OCTHER

Surface forcing:

If $ISOLPR \neq 0$, the solar radiation I_o is allowed to penetrate *NSOLPEN* layers downward on the ice-free part of each grid cell. The penetration is described by a simple vertical profile *I*, constant with latitude and longitude

$$\frac{I}{I_o} = (1-R)e^{z/D} \cdot (1-A_I), \qquad (4.1.1)$$

which is calculated from the parameters set in subroutine BODEN (see Paulson and Simpson (1977) for reference). A_I is the sea-ice concentration.

The radiation profile *I* is converted into an absorption profile I^a and this is used to update the oceanic temperature of the upper layers:

$$\theta_k^{n+\frac{l+1}{m}} = \theta_k^{n+\frac{l}{m}} + \frac{\Delta t}{\rho_o c_p d_{wk}} \cdot I_k^a, \qquad k=1,NSOLPEN. \tag{4.1.2}$$

In shallow areas, the profile is adjusted in order to guarantee conservation of heat in each water column.

If $ICESEA \neq 1$, i.e. the sea-ice model is not active, the first layer temperature is updated by a combination of a net atmospheric heat flux Q_o, a heat-flux correction term Q_Δ, and a relaxation term to observed SSTs . The weights w_r are set in subroutine COUPINI. Positive fluxes are upward:

$$\theta_1^{n+\frac{l+1}{m}} = \theta_1^{n+\frac{l}{m}} -$$

$$\frac{\Delta t}{\rho_o c_p (\Delta z_{w1} + \zeta^n)} (Q_0 + Q_\Delta) \cdot w_r + \Delta t \lambda_\theta \left(\theta_{obs} - \theta_1^{n+\frac{l}{m}} \right) \cdot (1 - w_r) . \tag{4.1.3}$$

If $IPME \neq 0$ and $ICESEA \neq 1$, a freshwater flux $\Delta\zeta = \Delta t \cdot (P - E)$ is added to the surface elevation and the surface salinity is updated accordingly:

$$\zeta^{n+\frac{l+1}{m}} = \zeta^{n+\frac{l}{m}} + \Delta\zeta \tag{4.1.4}$$

$$S_1^{n+\frac{l+1}{m}} = S_1^{n+\frac{l}{m}} \left(\Delta z_{w1} + \zeta^{n+\frac{l}{m}} \right) \Big/ \left(\Delta\zeta + \Delta z_{w1} + \zeta^{n+\frac{l}{m}} \right). \tag{4.1.5}$$

$(P - E)$ is the freshwater flux rate.

If $ICESEA = 1$, i.e. the sea-ice model is activated, the update by atmospheric heat and freshwater fluxes is done in subroutine OCICE. New ice velocities and compactness are calculated there as well.

If $IRELSAL > 0$, the surface salinity is relaxed to an observed SSS field and a corresponding freshwater flux is added to the surface elevation:

$$S_1^{n+\frac{l+1}{m}} = S_1^{n+\frac{l}{m}} + \Delta t \lambda_S \left(S_{obs} - S_1^{n+\frac{l}{m}} \right) \tag{4.1.6}$$

$$\zeta^{n+\frac{l+1}{m}} = \zeta^{n+\frac{l}{m}} - \Delta t \lambda_S \left(S_{obs} - S_1^{n+\frac{l}{m}} \right) \left(\Delta z_{w1} + \zeta^{n+\frac{l}{m}} \right) / S^{n+\frac{l+1}{m}} \; . \tag{4.1.7}$$

If $IRUNO \neq 0$, continental runoff is added to the surface elevation and the surface salinity is updated accordingly (see (4.1.5) and (4.1.4)).

Baroclinic pressure and stability:

From the last update of potential temperatures and salinity, $\theta_k^{n+l/m}$ and $S_k^{n+l/m}$, density, hydrostatic pressure, and stability are computed. First, potential temperatures θ are converted into in-situ temperatures T in subroutine ADISIT. The relative density ρ / ρ_o is calculated in subroutine RHO1 using the UNESCO equation of state.
The internal pressure is calculated with horizontally constant reference pressure $p_k^{ref} = \rho_o g h_{uk}$ where h_{uk} *(=TIESTU(k)>0)* is the depth of layer k:

$$p_1 = g \Delta z_{w1} \rho(T_1^{n+l/m}, S_1^{n+l/m}, p_1^{ref}) \tag{4.1.8}$$

$$p_k = p_{k-1} + g \Delta z_{wk} \rho(T_k^{n+l/m}, S_k^{n+l/m}, p_k^{ref}) \, . \tag{4.1.9}$$

A stability criterion (stored in array $STABI_{O/E}$) for later convective adjustment is calculated from vertical differences of density with the θ, S characteristics of the layers $k-1$ and k, evaluated at the depth of layer k:

$$\frac{\partial \rho'_k}{\partial z} = \frac{\rho(T_k^{n+l/m}, S_k^{n+l/m}, p_k^{ref}) - \rho(T_{k-1}^{n+l/m}, S_{k-1}^{n+l/m}, p_k^{ref})}{\rho_o \Delta z_{uk}} \, . \tag{4.1.10}$$

Convective adjustment:

Convective adjustment is activated for the parameter setting *ICONVA = 1*.
It is done by mixing of each pair of overlying unstable grid cells, with only one sweep through the water column per time step, i.e. for water cells with $\partial\rho'_k/\partial z < -DSTAB$, temperature and salinity are updated by:

$$\theta_{k-1}^{n+\frac{l+1}{m}} = \theta_k^{n+\frac{l+1}{m}} = \frac{d_{w(k-1)}\theta_{k-1}^{n+\frac{l}{m}} + d_{wk}\theta_k^{n+\frac{l}{m}}}{d_{w(k-1)} + d_{wk}} \tag{4.1.11}$$

$$S_{k-1}^{n+\frac{l+1}{m}} = S_k^{n+\frac{l+1}{m}} = \frac{d_{w(k-1)}S_{k-1}^{n+\frac{\dot{l}}{m}} + d_{wk}S_k^{n+\frac{l}{m}}}{d_{w(k-1)} + d_{wk}}. \tag{4.1.12}$$

$DSTAB \geq 0$ has to be specified (usually = *0*). If convective adjustment applies, the stratification array $STABI_{O/E}$ is set to zero.The surface elevation and ice draft is accounted for in d_{w1}.

For stable stratification, i.e. $\partial\rho'_k/\partial z \geq 0$ or if *ICONVA = 0*, θ and S remain unchanged:

$$\theta_k^{n+\frac{l+1}{m}} = \theta_k^{n+\frac{l}{m}} \tag{4.1.13}$$

$$S_k^{n+\frac{l+1}{m}} = S_k^{n+\frac{l}{m}}. \tag{4.1.14}$$

Richardson-number dependent vertical diffusivity/viscosity:

For the parameter setting $IEDDY = 1$, Richardson-number dependent coefficients for vertical eddy viscosity, A_V, and eddy diffusivity, D_V, are computed every timestep (Pacanowski and Philander, 1981). A mixed-layer turbulence contribution and constant background turbulence, A_b, and viscosity, D_b, are also included:

$$A_V^{n+1} = (1-\lambda_V)A_V^n + \lambda_V\left(\frac{A_{V0}}{(1 + C_{RA}Ri)^2} + \delta_{\Delta\rho}W_\rho + A_b\right) \tag{4.1.15}$$

$$D_V^{n+1} = (1-\lambda_V)D_V^n + \lambda_V Max[\left(\frac{D_{V0}}{(1 + C_{RD}Ri)^3} + \delta_{\Delta\rho}W_\rho\right), D_b], \tag{4.1.16}$$

with $0 \leq \lambda_V \leq 1$. A_{V0}, D_{V0}, C_{RA} and C_{RD} are constant values. λ_V defines a memory time scale.

Ri is the local Richardson number (see (4.1.10)):

$$Ri = \frac{g\partial\rho'/\partial z}{(\partial u/\partial z)^2 + (\partial v/\partial z)^2} \quad . \tag{4.1.17}$$

The term $\delta_{\Delta\rho} W_\rho$ is a mixed-layer turbulence term which is switched on for weak stratification ($\delta_{\Delta\rho} = 1$ if the vertical difference with the sea-surface density is smaller than a specified $\Delta\rho$, and 0 otherwise). W_ρ decreases rapidly with depth.

For *IEDDY = 0* , the eddy coefficients A_V and D_V are constant in time and can be specified to vary with water depth.

4.2 Subroutine OCWIND

Subroutine OCWIND updates the upper-layer ocean velocity by applying an atmosphere-ocean and ice-ocean surface stress:

$$\vec{v}_1^{\,n+\frac{l+1}{m}} = \vec{v}_1^{\,n+\frac{l}{m}} + \Delta t \frac{\vec{\tau}_a}{\Delta z_{w1} \cdot \rho_o}(1 - A_I) + \Delta t \frac{\vec{\tau}_I}{\Delta z_{w1} \cdot \rho_o} A_I \, . \tag{4.2.1}$$

A_I is the compactness of the sea-ice cover, $\vec{\tau}_I = \rho_o c_w \left|\vec{v}_1^{\,n+l/m} - \vec{v}_I^{\,n+1}\right| (\vec{v}_I^{\,n+1} - \vec{v}_1^{\,n+l/m})$ the ice-water stress with ice velocity $\vec{v}_I$, and $\vec{\tau}_a$ the wind stress. Note that the ice velocities of the actual time step are already calcultaed in OCICE (called in OCTHER) and that *l = 0* here.

4.3 Subroutine OCVISC

In subroutine OCVISC, horizontal velocities are modified due to 3-D eddy viscosity, vertical advection of momentum, and lateral and bottom friction.

The update by vertical advection of horizontal momentum and vertical eddy viscosity

$$\frac{\partial}{\partial t}(\vec{v} d_u) = \frac{\partial}{\partial z}\left(A_V \frac{\partial}{\partial z}(\vec{v} d_u) \right) - w \frac{\partial}{\partial z}(\vec{v} d_u) \tag{4.3.1}$$

is solved implicitly.

For the advection term, an upstream scheme is used with vertical velocities of the last timestep:

$$
\begin{aligned}
d_{uk}\dot{\vartheta}_k^{\,n+\frac{l+1}{m}} &= d_{uk}\dot{\vartheta}_k^{\,n+\frac{l}{m}} + \\
\Delta t\left[\frac{A_{Vk}\left(\frac{d_{u(k-1)}\dot{\vartheta}_{k-1}-d_{uk}\dot{\vartheta}_k}{\Delta z_{uk}}\right)^{n+\frac{l+1}{m}} - A_{V(k+1)}\left(\frac{d_{uk}\dot{\vartheta}_k-d_{u(k+1)}\dot{\vartheta}_{k+1}}{\Delta z_{u(k+1)}}\right)^{n+\frac{l+1}{m}}}{\Delta z_{wk}}\right] & \qquad (4.3.2)\\
-\Delta t\left[\left(-w_k^*\left(\frac{d_{u(k-1)}\dot{\vartheta}_{k-1}-d_{uk}\dot{\vartheta}_k}{\Delta z_{uk}}\right)^{n+\frac{l+1}{m}}\right)+w_{k+1}^*\left(\frac{d_{uk}\dot{\vartheta}_k-d_{u(k+1)}\dot{\vartheta}_{k+1}}{\Delta z_{u(k+1)}}\right)^{n+\frac{l+1}{m}}\right]
\end{aligned}
$$

where

$$\bar{w}_k = \frac{1}{4}(w_k^N + w_k^S + w_k^W + w_k^E)^n ,$$

$$w_k^* = \frac{1}{2}(|\bar{w}_k| - \bar{w}_k) ,$$

$$w_{k+1}^* = \frac{1}{2}(|\bar{w}_{k+1}| + \bar{w}_{k+1}) .$$

Superscripts *N,S,W,E* indicate the northern, southern etc. neighbours of the vector point in question. Equation (4.3.2) is solved by Gauss elimination with backsubstitution.

The horizontal viscous dissipation of momentum depends on the local rate of strain (see equation (2.4)). The Laplace operator is rotated by 45^o (see Figure 3; index *C* is omitted in equation (4.3.3)). The formulation conserves momentum in the absence of rigid boundaries. ν_A can be specified to be latitude dependent.

$$
\begin{aligned}
d_u\dot{\vartheta}^{n+(l+1)/m} &= d_u\dot{\vartheta}^{n+l/m} \\
&+ \Delta t\frac{\nu_A}{\Delta x\Delta y}[-(T_N^2+T_E^2)^n\,[d_u\dot{\vartheta}^{n+l/m}-(d_u\dot{\vartheta}^{n+l/m})_{NE}] \\
&\quad -(T_S^2+T_W^2)^n\,[d_u\dot{\vartheta}^{n+l/m}-(d_u\dot{\vartheta}^{n+l/m})_{SW}] \\
&\quad -(T_N^2+T_W^2)^n[d_u\dot{\vartheta}^{n+l/m}-(d_u\dot{\vartheta}^{n+l/m})_{NW}] \\
&\quad -(T_S^2+T_E^2)^n\,[d_u\dot{\vartheta}^{n+l/m}-(d_u\dot{\vartheta}^{n+l/m})_{SE}]\quad]
\end{aligned}
\qquad (4.3.3)
$$

where

$$(T^2)^n = \lambda_{T1}\left(\frac{\partial v^n}{\partial x} + \frac{\partial u^n}{\partial y}\right)^2 + \lambda_{T2}(T^2)^{n-1} \tag{4.3.4}$$

and $0 \le \lambda_{T1}, \lambda_{T2} \le 1$. T^2 is calculated in subroutine OCTURB which is called in OCVTOT. $\lambda_{T1}, \lambda_{T2}$ define a memory time scale.

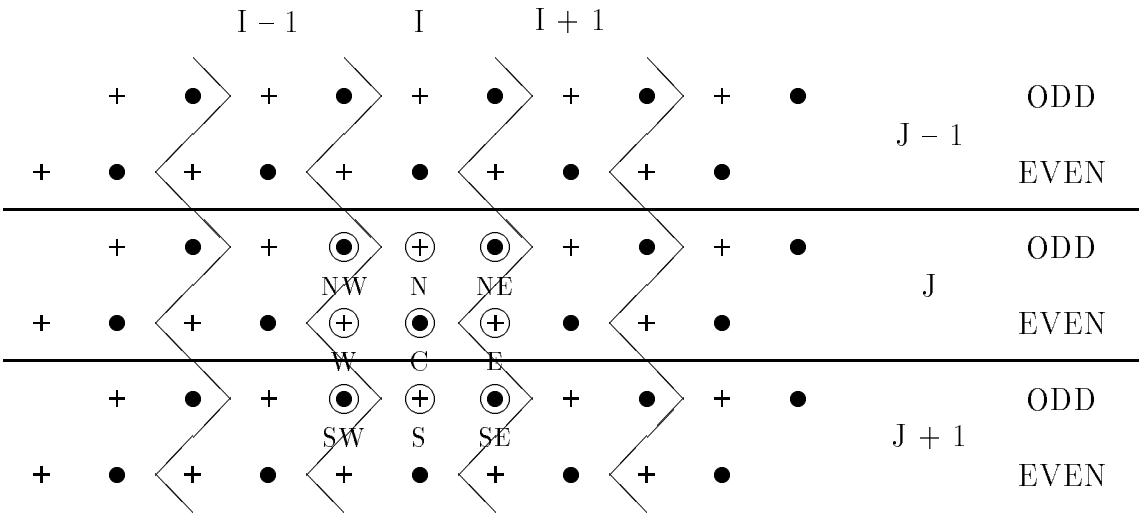


Figure 3 **Gridpoints used for the strain-rate dependent viscosity operator**

The harmonic dissipation tensor with constant coefficient (see equation (2.2)) also conserves momentum:

$$d_u\hat{v}^{n+(l+1)/m} = d_u\hat{v}^{n+l/m}$$

$$+ \frac{\Delta t}{\Delta x}\cdot\left(\left(d_u\hat{v}^{n+\frac{l}{m}}\right)_{WW} - d_u\hat{v}^{n+\frac{l}{m}}\right)\cdot A_H/\Delta x_W$$

$$+ \frac{\Delta t}{\Delta x}\cdot\left(\left(d_u\hat{v}^{n+\frac{l}{m}}\right)_{EE} - d_u\hat{v}^{n+\frac{l}{m}}\right)\cdot A_H/\Delta x_E$$

$$+ \frac{\Delta t}{\Delta x\Delta y}\cdot\left(\left(d_u\hat{v}^{n+\frac{l}{m}}\right)_{NN} - d_u\hat{v}^{n+\frac{l}{m}}\right)\cdot A_H^N\cdot\frac{\Delta x_N}{\Delta y_N}$$

$$+\frac{\Delta t}{\Delta x \Delta y}\cdot\left(\left(d_u \hat{v}^{\,n+\frac{l}{m}}\right)_{SS} - d_u \hat{v}^{\,n+\frac{l}{m}}\right)\cdot A_H^S \cdot \frac{\Delta x_S}{\Delta y_S} \quad . \tag{4.3.5}$$

The coefficient A_H can be latitude-dependent. For the indexing see Figure 4. Index C is omitted in equation (4.3.5).

Lateral friction is linear and solved implicitly:

$$\frac{\partial v}{\partial t} = -|f|\cdot\delta(\partial)\cdot\hat{v} \tag{4.3.6}$$

$$\hat{v}^{n+(l+1)/m} = \hat{v}^{n+l/m} - \Delta t\cdot|f|\cdot\delta(\partial)\hat{v}^{n+(l+1)/m} \quad . \tag{4.3.7}$$

with $\delta(\partial) = 1$ if a horizontally neighbouring cell is dry and 0 otherwise.

Bottom friction is linear as well:

$$\frac{\partial}{\partial t}\hat{v} = -\varepsilon\cdot\delta(-H)\cdot\hat{v} \tag{4.3.8}$$

$$\hat{v}^{n+(l+1)/m} = \hat{v}^{n+l/m} - \Delta t\cdot\varepsilon\cdot\delta(-H)\hat{v}^{n+l/m} \quad . \tag{4.3.9}$$

with $\delta(-H) = 1$ for the deepest cell of each water column and 0 elsewhere.

4.4 Subroutine OCBIHAE

The subroutine OCBIHAE can be used to reduce grid separation.

The biharmonic operator in

$$\frac{\partial}{\partial t}\hat{v} = -B_{HR}\nabla^4(\hat{v}) \tag{4.4.1}$$

is rotated by 45^o (as opposed to subroutine OCBIHAR), whereby exchanges between odd and even rows occur. For the spatial discretisation see e.g. Abramowitz and Stegun, 1972, page 885. The resulting diffusion is grid-size dependent (Δx is constant):

$$\hat{v}^{n+(l+1)/m} = \hat{v}^{n+l/m} - \frac{\Delta t B_{HR}}{(\Delta x)^4}[20\hat{v} - 8(\hat{v}_{NE} + \hat{v}_{SE} + \hat{v}_{NW} + \hat{v}_{SW}) + 2(\hat{v}_{NN} + \hat{v}_{SS} + \hat{v}_{WW} + \hat{v}_{EE}) + (\hat{v}_{NNEE} + \hat{v}_{SSEE} + \hat{v}_{NNWW} + \hat{v}_{SSWW})]^{n+l/m} \quad (4.4.2)$$

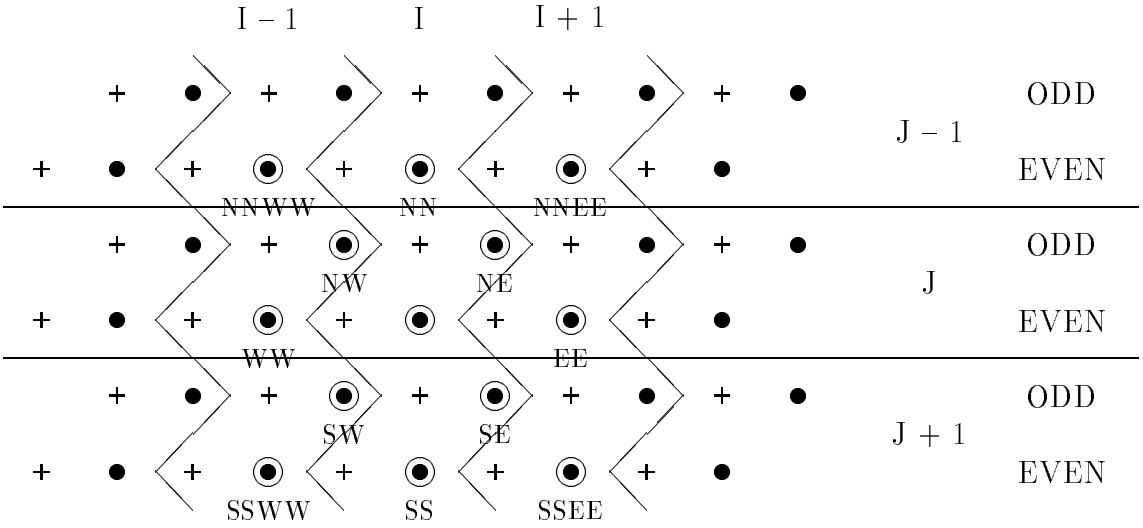


Figure 4 **Gridpoints used for the rotated biharmonic viscosity operator**

4.5 Subroutine OCBIHAR

In contrast to subroutine OCBIHAE, vector points from even and odd rows do not exchange with each other with the unrotated biharmonic operator

$$\frac{\partial}{\partial t}\hat{v} = -B_H \nabla^4(\hat{v}) \quad (4.5.1)$$

applied here:

$$\hat{v}^{n+(l+1)/m} = \hat{v}^{n+l/m} - \frac{\Delta t B_H}{(\Delta x)^4}[20\hat{v} - 8(\hat{v}_{NN} + \hat{v}_{SS} + \hat{v}_{WW} + \hat{v}_{EE}) + 2(\hat{v}_{NNEE} + \hat{v}_{SSEE} + \hat{v}_{NNWW} + \hat{v}_{SSWW}) + (\hat{v}_{NNNN} + \hat{v}_{SSSS} + \hat{v}_{WWWW} + \hat{v}_{EEEE})]^{n+l/m} \quad (4.5.2)$$

As is the formulation of the rotated biharmonic friction, the effective viscosity is grid-size dependent and not momentum conserving since Δx is constant. For the indexing see Figure 5.

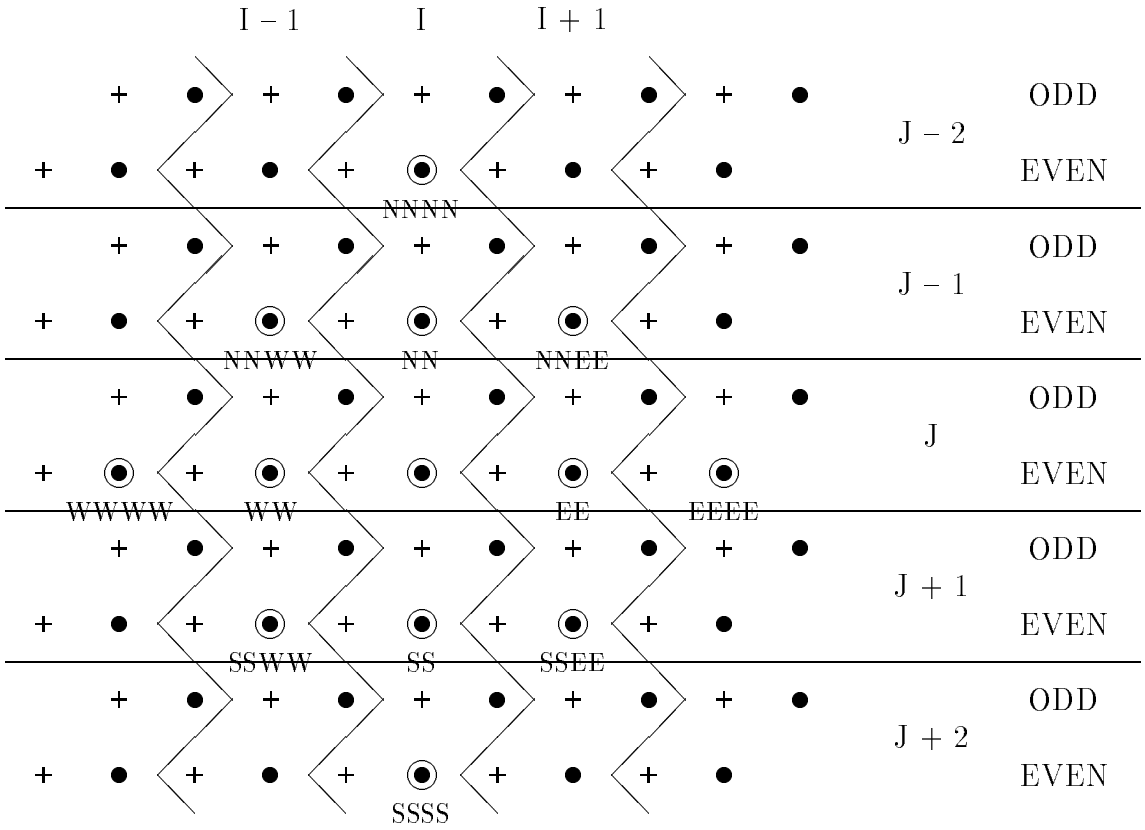


Figure 5 **Gridpoints used for the non-rotated biharmonic viscosity operator**

4.6 Subroutine OCIADJ7

Horizontal momentum advection

$$\frac{\partial \gamma}{\partial t} + u\frac{\partial \gamma}{\partial x} + v\frac{\partial \gamma}{\partial y} = 0 \tag{4.6.1}$$

(for $\gamma = u, v$) is done with a spatial discretisation scheme which conserves kinetic energy and enstrophy for non-divergent flow (Operator J7 of Arakawa and Lamb, 1977).

The time discretisation is done with a half-split method:

$$\gamma^{*} = \gamma^{n+\frac{l}{m}} - \frac{\Delta t}{2}\left(u^{n+\frac{l}{m}} \frac{\partial \gamma}{\partial x}^{n+\frac{l}{m}} + v^{n+\frac{l}{m}} \frac{\partial \gamma}{\partial y}^{n+\frac{l}{m}} \right) \tag{4.6.2}$$

$$\gamma^{n+\frac{l+1}{m}} = \gamma^{n+\frac{l}{m}} - \Delta t\left(u^{n+\frac{l}{m}} \frac{\partial \gamma}{\partial x}^{*} + v^{n+\frac{l}{m}} \frac{\partial \gamma}{\partial y}^{*} \right). \tag{4.6.3}$$

4.7 Subroutine OCMODS

The velocity fields $\vec{v} = (u, v)$ are decomposed into a barotropic and a baroclinic part:

$$\vec{V} = \int_{-H}^{0} \vec{v}\, dz \tag{4.7.1}$$

$$\vec{v}' = \vec{v} - \frac{\vec{V}}{H}. \tag{4.7.2}$$

4.8 Subroutine OCCLIT GEOPOL

The baroclinic subsystem is solved here (see “Baroclinic subsystem” on page 29 for a detailed description). Near the pole, the baroclinic velocities are assumed to be geostrophic.

4.9 Subroutine OCRHSZ

The right-hand side of the barotropic system is computed here (see “Barotropic subsystem” on page 33 for a detailed description).

4.10 Subroutine ZGAUSS

The barotropic subsystem is solved here directly by Gauss elimination (see “Direct solution of the barotropic system” on page 38) with the barotropic system matrix read from disk.

4.11 Subroutine ZGAUSS2

The barotropic subsystem is solved here directly by Gauss elimination (see “Direct solution of the barotropic system” on page 38) with the barotropic system matrix kept in core memory.

4.12 Subroutine ZSOR

The barotropic subsystem is solved here iteratively (see “Iterative solution of the barotropic system” on page 41).

4.13 Subroutine OCVTRO

The “new” sea-level values are used to calculate new barotropic velocities $\vec{V}^{n+1}$ using equations (6.1.7).

4.14 Subroutine OCVTOT

Baroclinic and barotropic velocities are combined to give the total velocity fields $\vec{v}^{n+1}$.

The vertical velocity is calculated from the continuity equation

$$w^{n+1}\big|_{z'=z} = -\int_{-H}^{z} \beta\nabla(\vec{v}^{n+1})dz' - \int_{-H}^{z} (1-\beta)\nabla(\vec{v}^{n})dz', \qquad (4.14.1)$$

where β is a weighting factor for the new and the old time levels (see “Baroclinic subsystem” on page 29).

A test for the correct definition of the barotropic implicit system and/or the correct backsubstitution is available at this point, i.e. the consistency of sea-level change with the vertical velocity at $z = 0$

$$\frac{\partial\zeta}{\partial t} - w\big|_{z=0} = 0 \qquad (4.14.2)$$

can be checked here.
The result of this test should be of the order of round-off-errors (modified by the effects of twofold differencing, empirically: 10^{-13} on a CRAY C90).

The time levels of velocity and sea level fields are updated here.

Finally, subroutine OCTURB is called, where the velocity-shear dependent coefficients T^2, used in the shear-dependent viscosity and diffusivity operators, arecalculated.

4.15 Subroutine OCADUP

Subroutine OCADUP is called when $ITSADV = 0$. This subroutine updates the potential temperature and salinity through advection using an upstream scheme, and through harmonic horizontal diffusion $(\gamma = \theta, S)$:

$$\frac{\partial\gamma}{\partial t} = -u\frac{\partial\gamma}{\partial x} - v\frac{\partial\gamma}{\partial y} - w\frac{\partial\gamma}{\partial z} + D_H\nabla^2(\gamma)\,. \qquad (4.15.1)$$

4.16 Subroutine OCADV

Subroutine OCADV is called when $ITSADV \neq 0$. This subroutine updates potential temperature and salinity through advection using a centered scheme in space. It also adds harmonic horizontal diffusion with constant coefficient and, if $IHMIX = 1$, shear-dependent diffusion:

$$\frac{\partial \gamma}{\partial t} = -u\frac{\partial \gamma}{\partial x} - v\frac{\partial \gamma}{\partial y} - w\frac{\partial \gamma}{\partial z} + D_H \nabla^2(\gamma) + \nabla(\nu_D T^2 \vec{\nabla}_H(\gamma)) \tag{4.16.1}$$

for $\gamma = \theta, S$.
For the discretisation in time, a predictor-corrector scheme is used:

$$\gamma^* = \gamma^{n+l/m} - \Delta t\left(u^{n+1}\frac{\partial \gamma}{\partial x}^{n+l/m} + v^{n+1}\frac{\partial \gamma}{\partial y}^{n+l/m} + w^{n+1}\frac{\partial \gamma}{\partial z}^{n+l/m}\right) \tag{4.16.2}$$

$$\begin{aligned}\gamma^{n+(l+1)/m} = \gamma^{n+l/m} &- \Delta t\left(u^{n+1}\frac{\partial \gamma}{\partial x}^{*} + v^{n+1}\frac{\partial \gamma}{\partial y}^{*} + w^{n+1}\frac{\partial \gamma}{\partial z}^{*}\right) \\ &+ \Delta t \cdot (D_H \nabla^2(\gamma^*) + \nabla(\nu_D T^2 \vec{\nabla}_H(\gamma^*))) \quad .\end{aligned} \tag{4.16.3}$$

4.17 Subroutine OCVERDI

Subroutine OCVERDI updates the thermohaline fields $(\gamma = \theta, S)$ through vertical eddy diffusion:

$$\frac{\partial \gamma}{\partial t} = \frac{\partial}{\partial z}\left(D_V \frac{\partial \gamma}{\partial z}\right). \tag{4.17.1}$$

The time discretisation is implicit:

$$\gamma_k^{n+(l+1)/m} = \gamma_k^{n+l/m} + \Delta t\left[\frac{D_{Vk}\left(\frac{\gamma_{k-1}-\gamma_k}{\Delta z_{uk}}\right)^{n+(l+1)/m} - D_{V(k+1)}\left(\frac{\gamma_k-\gamma_{k+1}}{\Delta z_{u(k+1)}}\right)^{n+(l+1)/m}}{\Delta z_{wk}}\right] \tag{4.17.2}$$

Equation (4.17.2) is solved by Gauss elimination and backsubstitution. The Richardson-number dependent vertical diffusion coefficient D_V is computed in subroutine OCTHER.

4.18 Subroutine OCPODI

Subroutine OCPODI is called if parameter *NPOLE* is set to *1*. The scalar grid points nearest to the North Pole are land points in each subgrid (*J=1,2*; Even/Odd). The thermohaline fields of the next latitudes in each subgrid (*J=JGEOO,JGEOU*) are replaced by their zonal means. Thus, zonal pressure gradients near the polar solid wall are eliminated. If the grid does not contain the North Pole, this subroutine must not be called (set *NPOLE = 0*).

5 Baroclinic subsystem

For the baroclinic subsystem, it is assumed that disturbances are of small amplitude, i.e. variations in ρ can be approximated by $\partial\rho/\partial t + w\partial\rho/\partial z = 0$. Thus, only the linear wave aspect of the full baroclinic system is considered. The behaviour of these disturbances dominates the stability behaviour of the full system.

The baroclinic subsystem is described by the baroclinic momentum equations

$$\frac{\partial}{\partial t}u' - fv' = -\left(\frac{\partial}{\partial x}p' - \frac{1}{H}\int_{-H}^{0}\frac{\partial}{\partial x}p'dz\right) + G'_u, \tag{5.1}$$

$$\frac{\partial}{\partial t}v' + fu' = -\left(\frac{\partial}{\partial y}p' - \frac{1}{H}\int_{-H}^{0}\frac{\partial}{\partial y}p'dz\right) + G'_v, \tag{5.2}$$

an equation for the time evolution of the internal pressure

$$\frac{\partial^2}{\partial z\partial t}p' = \frac{wg}{\rho_o}\frac{\partial\rho}{\partial z}, \tag{5.3}$$

and the baroclinic continuity equation

$$\frac{\partial}{\partial z}w' = -\nabla\hat{v}'. \tag{5.4}$$

Equations (5.1), (5.2), and (5.4) are obtained by subtracting the vertical mean of the momentum and continuity equations from these equations respectively (see section 4.7). p' is the internal pressure divided by the reference density ρ_o. G'_u and G'_v are the deviations of the friction and advection terms in equation (2.1) from the vertical mean.

To obtain equation (5.3), the small-amplitude approximation of the density advection equation $\partial\rho/\partial t = -w\partial\rho/\partial z$ is combined with the time derivative of the hydrostatic equation $\partial p'^2/\partial z\partial t = -g/\rho_o\partial\rho/\partial t$.

5.1 Time discretisation

In the following, the "old" values of the variables (with superscript n) contain all update informations that are available according to the calling order of the subroutines in OCESTEP as described in section 4. Subscripts x, y, z, and t denote partial derivatives in space and time.
The time-stepping scheme of the Coriolis and the pressure terms can be modified by the choice of factors α and β, with $0 \le \alpha \le 1$ and $0 \le \beta \le 1$, which determine the relative weights of the variables at the new and old time levels. Note that at the time the baroclinic subsystem is solved, the velocity is already updated by the viscosity and momentum advection operators. The baroclinic equations thus become:

$$u'^{n+1} - u'^{n} = \alpha\Delta t\left(f v'^{n+1} - p'^{n+1}_{x} + \frac{1}{H}\int_{-H}^{0} p'^{n+1}_{x}\,dz \right) + (1-\alpha)\Delta t\left(f v'^{n} - p'^{n}_{x} + \frac{1}{H}\int_{-H}^{0} p'^{n}_{x}\,dz \right) \tag{5.1.1}$$

$$v'^{n+1} - v'^{n} = \alpha\Delta t\left(-f u'^{n+1} - p'^{n+1}_{y} + \frac{1}{H}\int_{-H}^{0} p'^{n+1}_{y}\,dz \right) + (1-\alpha)\Delta t\left(-f u'^{n} - p'^{n}_{y} + \frac{1}{H}\int_{-H}^{0} p'^{n}_{y}\,dz \right) \tag{5.1.2}$$

$$p'^{n+1}_{z} - p'^{n}_{z} = \frac{g}{\rho_o}\Delta t \rho_z (\beta w'^{n+1} + (1-\beta) w'^{n}) . \tag{5.1.3}$$

The influence of α and β on wave-propagation characteristics and system stability is described in section "Stability of linear free waves" on page 47.

Equations (5.1.1) and (5.1.2) are reorganized to give direct expressions for u'^{n+1} and v'^{n+1}:

$$u'^{n+1} - f\alpha\Delta t v'^{n+1} = \Gamma_u(\vec{v}'^{n}, p'^{n}) + \Pi_u^{n+1}(p'^{n+1}) \tag{5.1.4}$$

$$v'^{n+1} + f\alpha\Delta t u'^{n+1} = \Gamma_v(\vec{v}'^{n}, p'^{n}) + \Pi_v^{n+1}(p'^{n+1}) \tag{5.1.5}$$

where

$$\Gamma_u = u'^n + (1-\alpha)\Delta t\left(f v'^n - p'^n_x + \frac{1}{H}\int_{-H}^{0} p'^n_x dz \right) \tag{5.1.6}$$

$$\Gamma_v = v'^n + (1-\alpha)\Delta t\left(-f u'^n - p'^n_y + \frac{1}{H}\int_{-H}^{0} p'^n_y dz \right) \tag{5.1.7}$$

$$\Pi_u^{n+1} = -\alpha\Delta t\left(p'^{n+1}_x - \frac{1}{H}\int_{-H}^{0} p'^{n+1}_x dz \right) \tag{5.1.8}$$

$$\Pi_v^{n+1} = -\alpha\Delta t\left(p'^{n+1}_y - \frac{1}{H}\int_{-H}^{0} p'^{n+1}_y dz \right) \tag{5.1.9}$$

$$\Rightarrow \begin{bmatrix} 1 & -f\alpha\Delta t \\ f\alpha\Delta t & 1 \end{bmatrix} \cdot \begin{bmatrix} u'^{n+1} \\ v'^{n+1} \end{bmatrix} = \begin{bmatrix} \Gamma_u + \Pi_u^{n+1} \\ \Gamma_v + \Pi_v^{n+1} \end{bmatrix} \tag{5.1.10}$$

$$\Rightarrow \begin{bmatrix} u'^{n+1} \\ v'^{n+1} \end{bmatrix} = \frac{1}{1+(f\alpha\Delta t)^2} \begin{bmatrix} 1 & f\alpha\Delta t \\ -f\alpha\Delta t & 1 \end{bmatrix} \cdot \begin{bmatrix} \Gamma_u + \Pi_u^{n+1} \\ \Gamma_v + \Pi_v^{n+1} \end{bmatrix} \tag{5.1.11}$$

The system (5.1) - (5.4) is solved iteratively. The first guesses of Π_u^{n+1} and Π_v^{n+1} are Π_u^n and Π_v^n, respectively. The first guess of *w'* is computed from the old baroclinic velocities.
The iteration procedure is as follows (where l denotes the last iteration step):

1. Equation (5.1.11) is used to compute the next estimate of the baroclinic velocities

$$\Rightarrow \begin{bmatrix} u'^{n+1,l+1} \\ v'^{n+1,l+1} \end{bmatrix} = \frac{1}{1+(f\alpha\Delta t)^2} \begin{bmatrix} 1 & f\alpha\Delta t \\ -f\alpha\Delta t & 1 \end{bmatrix} \cdot \begin{bmatrix} \Gamma_u + \Pi_u^{n+1,l} \\ \Gamma_v + \Pi_v^{n+1,l} \end{bmatrix}$$

2. Equation (5.4) is used to update the vertical velocities

$$w'^{n+1,l+1}\Big|_{z'=z} = -\int_{-H}^{z} \nabla \cdot (u'^{n+1,l+1}, v'^{n+1,l+1}) dz'$$

3. Equation (5.1.3), (5.1.8), and (5.1.9) are then used to calculate the baroclinic pressure

$$p'^{n+1,l+1}_z = p'^{n}_z + \frac{g}{\rho_o}\Delta t \rho_z(\beta w'^{n+1,l+1} + (1-\beta)w'^{n})$$

$$\Pi_u^{n+1,l+1} = -a\Delta t\left(p'^{n+1,l+1}_x - \frac{1}{H}\int_{-H}^{0} p'^{n+1,l+1}_x dz\right)$$

$$\Pi_v^{n+1,l+1} = -a\Delta t\left(p'^{n+1,l+1}_y - \frac{1}{H}\int_{-H}^{0} p'^{n+1,l+1}_y dz\right)$$

4. Go back to step 1. or stop if the convergence criterion is reached (the number of iterations depends on the stability of the stratification and the time step).

6 Barotropic subsystem

The prognostic equation for the sea-surface elevation ζ is derived from the vertical integral of the momentum equations (see section 4.7):

$$\frac{\partial U}{\partial t} - fV + gH\frac{\partial \zeta}{\partial x} + \int_{-H}^{0} \frac{\partial}{\partial x} p' dz = G_U, \tag{6.1}$$

$$\frac{\partial V}{\partial t} + fU + gH\frac{\partial \zeta}{\partial y} + \int_{-H}^{0} \frac{\partial}{\partial y} p' dz = G_V, \tag{6.2}$$

$$\frac{\partial \zeta}{\partial t} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = Q_\zeta. \tag{6.3}$$

$p' = p/\rho_o$ denotes the baroclinic pressure divided by the constant reference density. (G_U,G_V) are the vertical integrals of the friction and advection terms in equation (2.1).

6.1 Time discretisation

According to the formulation of the baroclinic subsystem, the pressure and Coriolis terms are weighted averages of the terms at the old and new time levels. Note that at the time the barotropic subsystem is solved, the surface elevation is already updated by the freshwater fluxes, and that the velocity is updated by the viscosity and momentum advection operators. Subscripts x, y, z, and t denote partial derivatives in space and time. The barotropic momentum equations thus are:

$$U^{n+1} - U^n - f\Delta t(\alpha V^{n+1} + (1-\alpha)V^n) + gH\Delta t(\alpha \zeta_x^{n+1} + (1-\alpha)\zeta_x^n) + \Delta t \int_{-H}^{0} p'^{n+1}_x dz = 0 \tag{6.1.1}$$

$$V^{n+1} - V^n + f\Delta t(\alpha U^{n+1} + (1-\alpha)U^n) + gH\Delta t(\alpha \zeta_y^{n+1} + (1-\alpha)\zeta_y^n) + \Delta t \int_{-H}^{0} p'^{n+1}_y dz = 0 \tag{6.1.2}$$

$$\zeta^{n+1} - \zeta^n + \Delta t(\beta U_x^{n+1} + (1-\beta)U_x^n + \beta V_y^{n+1} + (1-\beta)V_y^n) = 0 \tag{6.1.3}$$

where α and β are defined as in section 5.

Equations (6.1.1) and (6.1.2) are reorganized to give direct expressions for U^{n+1} and V^{n+1}:

$$U^{n+1} - f\alpha\Delta t V^{n+1} = \Gamma_U(\vec{V}^n, \zeta^n, p'^{n+1}) + Z_U(\zeta^{n+1}) \tag{6.1.4}$$

$$V^{n+1} + f\alpha\Delta t U^{n+1} = \Gamma_V(\vec{V}^n, \zeta^n, p'^{n+1}) + Z_V(\zeta^{n+1}) \tag{6.1.5}$$

$$\Rightarrow \begin{bmatrix} 1 & -f\alpha\Delta t \\ f\alpha\Delta t & 1 \end{bmatrix} \cdot \begin{bmatrix} U^{n+1} \\ V^{n+1} \end{bmatrix} = \begin{bmatrix} \Gamma_U + Z_U \\ \Gamma_V + Z_V \end{bmatrix} \tag{6.1.6}$$

$$\Rightarrow \begin{bmatrix} U^{n+1} \\ V^{n+1} \end{bmatrix} = \frac{1}{1 + (f\alpha\Delta t)^2} \begin{bmatrix} 1 & f\alpha\Delta t \\ -f\alpha\Delta t & 1 \end{bmatrix} \cdot \begin{bmatrix} \Gamma_U + Z_U \\ \Gamma_V + Z_V \end{bmatrix} \tag{6.1.7}$$

where

$$\Gamma_U = U^n + f\Delta t(1-\alpha)V^n - gH\Delta t(1-\alpha)\zeta_x^n - \Delta t \int_{-H}^{0} p'^{n+1}_x dz \tag{6.1.8}$$

$$\Gamma_V = V^n - f\Delta t(1-\alpha)U^n - gH\Delta t(1-\alpha)\zeta_y^n - \Delta t \int_{-H}^{0} p'^{n+1}_y dz \tag{6.1.9}$$

$$Z_U = -gH\Delta t\alpha\zeta_x^{n+1} \tag{6.1.10}$$

$$Z_V = -gH\Delta t\alpha\zeta_y^{n+1}. \tag{6.1.11}$$

Equations (6.1.7) are now inserted into equation (6.1.3) to give an equation for ζ^{n+1} and its horizontal derivatives

$$\zeta^{n+1} + \Gamma_\zeta^{n+1} = \Gamma_\zeta^n \tag{6.1.12}$$

where

$$\Gamma_\zeta^{n+1} = -\alpha\beta g(\Delta t)^2 \Bigg(\frac{\partial}{\partial x}\left[\frac{1}{1 + (f\alpha\Delta t)^2}(H\zeta_x^{n+1} + f\alpha\Delta t H\zeta_y^{n+1})\right] + \frac{\partial}{\partial y}\left[\frac{1}{1 + (f\alpha\Delta t)^2}(H\zeta_y^{n+1} - f\alpha\Delta t H\zeta_x^{n+1})\right] \Bigg) \tag{6.1.13}$$

and

$$\Gamma_\zeta^n = \zeta^n - (1-\beta)\Delta t(U_x^n + V_y^n)$$

$$-\beta\Delta t\left(\frac{\partial}{\partial x}\left(\frac{1}{1+(f\alpha\Delta t)^2}(\Gamma_U + f\alpha\Delta t\Gamma_V)\right) + \frac{\partial}{\partial y}\left(\frac{1}{1+(f\alpha\Delta t)^2}(\Gamma_V - f\alpha\Delta t\Gamma_U)\right)\right)$$

$$\begin{aligned} = -\beta\Delta t\Bigg\{ & \frac{\partial}{\partial x}\Bigg(\frac{1}{1+(f\alpha\Delta t)^2}\Bigg[\langle U^n + f(1-\alpha)\Delta t V^n - gH(1-\alpha)\Delta t\zeta_x^n - \Delta t\int_{-H}^{0} p'^{n+1}_x dz\rangle \\ & + f\alpha\Delta t\langle V^n - f(1-\alpha)\Delta t U^n - gH(1-\alpha)\Delta t\zeta_y^n - \Delta t\int_{-H}^{0} p'^{n+1}_y dz\rangle\Bigg]\Bigg) \\ & + \frac{\partial}{\partial y}\Bigg(\frac{1}{1+(f\alpha\Delta t)^2}\Bigg[\langle V^n - f(1-\alpha)\Delta t U^n - gH(1-\alpha)\Delta t\zeta_y^n - \Delta t\int_{-H}^{0} p'^{n+1}_y dz\rangle \\ & - f\alpha\Delta t\langle U^n + f(1-\alpha)\Delta t V^n - gH(1-\alpha)\Delta t\zeta_x^n - \Delta t\int_{-H}^{0} p'^{n+1}_x dz\rangle\Bigg]\Bigg)\Bigg\} \\ & - (1-\beta)\Delta t(U_x^n + V_y^n) + \zeta^n \end{aligned} \tag{6.1.14}$$

6.2 Spatial discretisation

We next show how equation (6.1.12), the equation for the calculation of the sea-surface elevation, is discretized for even lines. The formulation involves new ζ-values at all points marked by open circles in Figure 6. All spatial differences are centered differences.

The grid distance between ζ_{NN} and ζ_C is labelled Δy_N, the distance between ζ_C and ζ_{SS} is labelled Δy_S, the distance between the northern and the southern vector point around ζ_C is Δy_C. The zonal distance between two ζ-points is labelled Δx_u and the zonal distance between two vector points is called Δx_ζ.

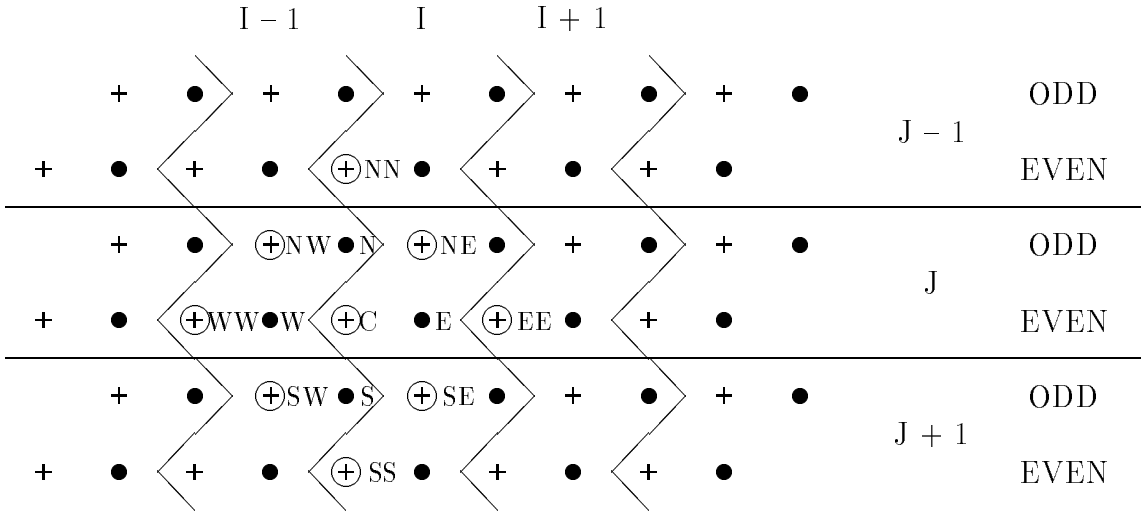


Figure 6 **Gridpoints involved in the calculation of new even sea-level values**

Equation (6.1.13) is discretized as follows (note the latitudinal variation of the Coriolis parameter and convergence of meridians; index *C* is omitted):

$$\Gamma_{\zeta}^{n+1} = \frac{-\alpha\beta g(\Delta t)^2}{\Delta x_{\zeta}}\left\{\frac{1}{1+(f\alpha\Delta t)^2}\left[\left(\frac{H_E(\zeta_{EE}^{n+1}-\zeta^{n+1})}{(\Delta x_u)_E}+f\alpha\Delta t\frac{H_E(\zeta_{NE}^{n+1}-\zeta_{SE}^{n+1})}{\Delta y}\right)\right.\right.$$
$$\left.\left.-\left(\frac{H_W(\zeta^{n+1}-\zeta_{WW}^{n+1})}{(\Delta x_u)_W}+f\alpha\Delta t\frac{H_W(\zeta_{NW}^{n+1}-\zeta_{SW}^{n+1})}{\Delta y}\right)\right]\right\}$$
$$\frac{-\alpha\beta g(\Delta t)^2}{\Delta y}\left\{\frac{1}{1+(f_N\alpha\Delta t)^2}\cdot\frac{(\Delta x_u)_N}{\Delta x_{\zeta}}\left[\frac{H_N(\zeta_{NN}^{n+1}-\zeta^{n+1})}{(\Delta y)_N}-f_N\alpha\Delta t\frac{H_N(\zeta_{NE}^{n+1}-\zeta_{NW}^{n+1})}{(\Delta x_u)_N}\right]\right.$$
$$\left.-\frac{1}{1+(f_S\alpha\Delta t)^2}\cdot\frac{(\Delta x_u)_S}{\Delta x_{\zeta}}\left[\frac{H_S(\zeta^{n+1}-\zeta_{SS}^{n+1})}{(\Delta y)_S}-f_S\alpha\Delta t\frac{H_S(\zeta_{SE}^{n+1}-\zeta_{SW}^{n+1})}{(\Delta x_u)_S}\right]\right\}$$

(6.2.1)

Equation (6.1.12) thus reads after collecting all coefficients and abbreviating $H^* = \frac{H}{1 + (f\alpha\Delta t)^2}$:

$$
\begin{aligned}
\Gamma_\zeta^n = \zeta^{n+1}\left[1 + \frac{\alpha\beta g(\Delta t)^2}{\Delta x_\zeta}\left(\frac{H_E^*}{(\Delta x_u)_E} + \frac{H_W^*}{(\Delta x_u)_W} + \frac{H_N^*(\Delta x_u)_N}{(\Delta y)_N \Delta y} + \frac{H_S^*(\Delta x_u)_S}{(\Delta y)_S \Delta y}\right)\right] \\
-\zeta_{EE}^{n+1}\left[\alpha\beta g(\Delta t)^2\left(\frac{H_E^*}{\Delta x_\zeta (\Delta x_u)_E}\right)\right] \\
-\zeta_{WW}^{n+1}\left[\alpha\beta g(\Delta t)^2\left(\frac{H_W^*}{\Delta x_\zeta (\Delta x_u)_W}\right)\right] \\
-\zeta_{NN}^{n+1}\left[\alpha\beta g(\Delta t)^2\left(\frac{H_N^*(\Delta x_u)_N}{\Delta x_\zeta (\Delta y)_N \Delta y}\right)\right] \\
-\zeta_{SS}^{n+1}\left[\alpha\beta g(\Delta t)^2\left(\frac{H_S^*(\Delta x_u)_S}{\Delta x_\zeta (\Delta y)_S \Delta y}\right)\right] \\
+\zeta_{NE}^{n+1}\left[\frac{\alpha^2\beta g(\Delta t)^3}{\Delta x_\zeta \Delta y}(f_N H_N^* - f_E H_E^*)\right] \\
+\zeta_{SE}^{n+1}\left[\frac{\alpha^2\beta g(\Delta t)^3}{\Delta x_\zeta \Delta y}(f_E H_E^* - f_S H_S^*)\right] \\
+\zeta_{NW}^{n+1}\left[\frac{\alpha^2\beta g(\Delta t)^3}{\Delta x_\zeta \Delta y}(f_W H_W^* - f_N H_N^*)\right] \\
+\zeta_{SW}^{n+1}\left[\frac{\alpha^2\beta g(\Delta t)^3}{\Delta x_\zeta \Delta y}(f_S H_S^* - f H_W^*)\right]
\end{aligned}
\tag{6.2.2}
$$

Equation (6.2.2) may be written as a matrix equation for all wet ocean points

$$
\underline{b} = A \cdot \underline{\zeta}^{n+1}, \tag{6.2.3}
$$

where A is the coefficient matrix of the right hand side of equation (6.2.2) and $\underline{b}$ the vector of the terms on the left hand side, with variables at the old time level only.

6.3 Direct solution of the barotropic system

Equation (6.2.3) can be solved directly with Gauss elimination and backsubstitution. In this case, the parameter *IELIMI* should be set to 1. If the coefficient matrix *A* fits in the core memory, this is the least time-consuming way to calculate new sea levels because the triangularisation has to be done only once at the beginning of the integration. The triangularisation is done in subroutine TRIAN and the backsubstitution in subroutine ZGAUSS2. Additional core memory requirements for A are $IMM \times ILL$ words, where IMM is twice the bandwidth *KBB* of the band diagonal sparse matrix A plus 1 for the central entry. ILL is the number of ocean scalar points in the upper-most layer.
If the barotropic system matrix does not fit in the core memory, the elimination factors and tridiagonal matrix entries can be read from a file, so that the full matrix is not core resident. The matrix has to be calculated then before the experiment starts. This situation creates an I/O burden during the integration, however. The number of I/O-operations per time step depends on the size of the buffer *(KMxIZGBUFF)* which is used to read the matrix. *IZGBUFF* should be specified as large as possible. The backsubstitution is done then in subroutine ZGAUSS.

6.3.1 Matrix triangularisation and backsubstitution

The oceanic scalar points are numbered consecutively in west-east direction and north-south direction. Each scalar point has its own row in the system matrix. The maximum number of entries in each row to be stored is the central entry and twice the bandwidth *KBB*.

The storing method and solution of the system is demonstrated in the following for a simple system matrix with one central and two neighbour points.

The coefficient matrix *A* of the equation

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & 0 & 0 & \dots \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & 0 & \dots \\ 0 & \alpha_{32} & \alpha_{33} & \alpha_{34} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \alpha_{N-1} & \alpha_{N-1,N-1} & \alpha_{N-1,N} \\ 0 & \dots & 0 & \alpha_{N,N-1} & \alpha_{N,N} \end{bmatrix} \cdot \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \dots \\ \zeta_{N-2} \\ \zeta_{N-1} \\ \zeta_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_{N-2} \\ b_{N-1} \\ b_N \end{bmatrix} \qquad (6.3.1.1)$$

is organized as follows (after division by the value of the main diagonal):

$$\begin{bmatrix} 0 & 1 & \alpha_{12}/\alpha_{11} \\ \alpha_{21}/\alpha_{22} & 1 & \alpha_{23}/\alpha_{22} \\ \alpha_{32}/\alpha_{33} & 1 & \alpha_{34}/\alpha_{33} \\ \dots & \dots & \dots \\ \alpha_{N-1,N-2}/\alpha_{N-1,N-1} & 1 & \alpha_{N-1,N}/\alpha_{N-1,N-1} \\ \alpha_{N,N-1}/\alpha_{N,N} & 1 & 0 \end{bmatrix} \tag{6.3.1.2}$$

Since in this simple example there are only two off-diagonal entries, only one entry in the lower diagonal part of A has to be eliminated per column. Triangularisation creates zero's in the first column of equation (6.3.1.2). For convenience the elimination factors are stored there. The elimination factors f_l (l runs down the column for more than one off diagonal entry) are then stored on the $(l-2)^{th}$ entry on row $l-1$. For the simple system this looks like

$$\begin{bmatrix} f_2 & 1 & \dfrac{\alpha_{12}}{\alpha_{11}} \\ f_3 & 1 - f_2\dfrac{\alpha_{12}}{\alpha_{11}} & \dfrac{\alpha_{23}}{\alpha_{22}} \\ f_4 & 1 - f_3\dfrac{\alpha_{23}}{\alpha_{22}} & \dfrac{\alpha_{34}}{\alpha_{33}} \\ \dots & \dots & \dots \\ f_N & 1 - f_{N-1}(\dots) & \dfrac{\alpha_{N-1,N}}{\alpha_{N-1,N-1}} \\ 0 & 1 - f_N\dfrac{\alpha_{N-1,N}}{\alpha_{N-1,N-1}} & 0 \end{bmatrix} \tag{6.3.1.3}$$

where

$$\begin{aligned} f_2 &= \frac{\alpha_{21}}{\alpha_{22}} \\ f_3 &= \frac{\alpha_{32}}{\alpha_{33}} \Big/ \left(1 - f_2\frac{\alpha_{12}}{\alpha_{11}}\right) \\ f_4 &= \frac{\alpha_{43}}{\alpha_{44}} \Big/ \left(1 - f_3\frac{\alpha_{23}}{\alpha_{22}}\right) \\ \dots &= \dots \\ f_N &= \dots \end{aligned}$$

In every time step the elimination factors f_l are used to manipulate the right hand side of equation (6.3.1.1) according to

$$
\begin{aligned}
b_1^* &= \frac{b_1}{\alpha_{11}} \\
b_2^* &= \frac{b_2}{\alpha_{22}} - f_2 b_1^* \\
b_3^* &= \frac{b_3}{\alpha_{33}} - f_3 b_2^* \\
\ldots &= \ldots \\
b_N^* &= \frac{b_N}{\alpha_{NN}} - f_N b_{N-1}^*.
\end{aligned}
$$

Backsubstitution is then done in the usual way without pivoting to keep the sparse band structure of the matrix. The maximum elimination factor must thus be checked when applying the model code to a new grid configuration. The system (6.2.2) basically contains the singularity of closed f/H contours. It may depend in an unforeseeable way on details of the chosen grid, whether the elimination develops detrimental accumulation of round off errors. A check of round-off errors is provided in subroutine OCVTOT.

6.3.2 Getting started with the direct solution

The direct solution is obtained with $IELIMI = 1$.
To calculate the dimension of the matrix, set $IMATW = 1$ and run HOPE. A check of the bandwidth KBB and the number of oceanic scalar points ILL in the upper-most layer is automatically done in HOPE.
The parameters ILL and KBB have to be adjusted to the values given in the printout:

```
TOTAL WET SCALAR POINTS IN LAYER 1 = ILL
FOR THE DIRECT SOLUTION SET ILL =....
BANDWIDTH = KBB
FOR THE DIRECT SOLUTION SET KBB =....
```

The bandwidth depends on the way of numbering the oceanic wet scalar points. The numbering should increase first in the direction of the smallest dimension of either IE or $2 \cdot JE$. For east-west periodic grids the numbering must always increase towards the east first.

If you want to keep the matrix core resident, all you have to do is to compile the model with subroutine TRIAN and ZGAUSS2 and to run it with $IMATW = -1$. The matrix is recalculated in subroutine TRIAN every time the model is started with $IMATW = -1$. This is done for safety reasons (see note below). Backsubstitution is done in subroutine ZGAUSS2.

If you want to read the matrix entries and elimination factors each time step, you have to create a file on which this information is stored. This still requires one short run with the matrix in core memory. The following three steps are necessary to run the model with the matrix on disk:

1. Compile the model with subroutines TRIAN and ZGAUSS.
   Set $IELIMI = 1$ and $IMATW = 1$ and run the model.
   Change ILL and KBB to the values given in the printout.

2. Recompile and rerun the model with $IELIMI = 1$ and $IMATW = 1$.
   This creates the matrix (subroutine TRIAN) and stores it on a file named *MATRIX* with the right choice of parameters *ILL* and *KBB*.

3. Recompile the model without subroutine TRIAN. Set $IELIMI = 1$ and $IMATW = 0$. This is the setting for integration with the matrix on disc. Backsubstitution is done then in subroutine ZGAUSS.

NOTE:
Every change in one of the terms in equation (6.2.3) on page 37 requires a new computation of the matrix (go again through steps 1 to 3). Any change in the following variables will change the matrix

$$\Delta t,\ f,\ \alpha,\ \beta,\ H,\ g,\ \Delta x,\ \Delta y$$

6.4 Iterative solution of the barotropic system

If the coefficient Matrix A does not fit in core memory and your computer center does not have a fast disk access (SSD) to read in the elimination factors from a file, the system (6.3.1.1) can be solved iteratively. In this case, the parameter $IELIMI$ should be set to zero.

The iteration method used in HOPE is referred to as “Successive Over-Relaxation (SOR) with Chebyshev acceleration” (see e.g. Press et al. 1986 and references therein).

Certain coefficient fields are preset in subroutine OCITPRE, and the iteration is done in subroutine ZSOR.

SOR is superior to other iterative procedures only in a very narrow window around the optimal ω.

How to choose ω?

If ρ_{Jacobi} is the spectral radius (the modulus of the amplitude of the slowest decaying eigenmode of the iteration matrix) of the Jacobi-iteration, then the optimal choice of ω is given by

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho_{Jacobi}^2}}$$

Although ω_{opt} is the optimal relaxation parameter, it is not necessarily a good initial choice. This is due to the fact that in SOR the error grows for the first iterations before asymptotic convergence sets in. A simple modification to SOR resolves this problem. In SOR with Chebyshev-acceleration ω is changed each half sweep in an odd-even configuration:

$$\omega_{E/O}^{(0)} = 1$$

$$\omega_{E/O}^{(n+1/2)} = \frac{1}{1 - \rho_{Jacobi}^2 \omega_{E/O}^{(n)} / 4} \qquad n = 0, \frac{1}{2}, 1, \ldots, \infty .$$

$\omega_{E/O}^{(n)}$ then converge to ω_{opt}.

6.4.1 Iteration procedure

The iteration is done on an 8-color grid to allow vectorisation (increment 1 in loops is essential for CYBER but recommended for CRAY machines too). Think of the 2-color chess board iteration for the solution of a Laplace equation. The west-east indices are grouped in two colors (array names ending on 1 or 2, e.g. Z_1 Z_2) and the north-south indices in four colors (third index K running from 1 to 4, e.g. $Z_1(I, J, K)$, see Figure 7).

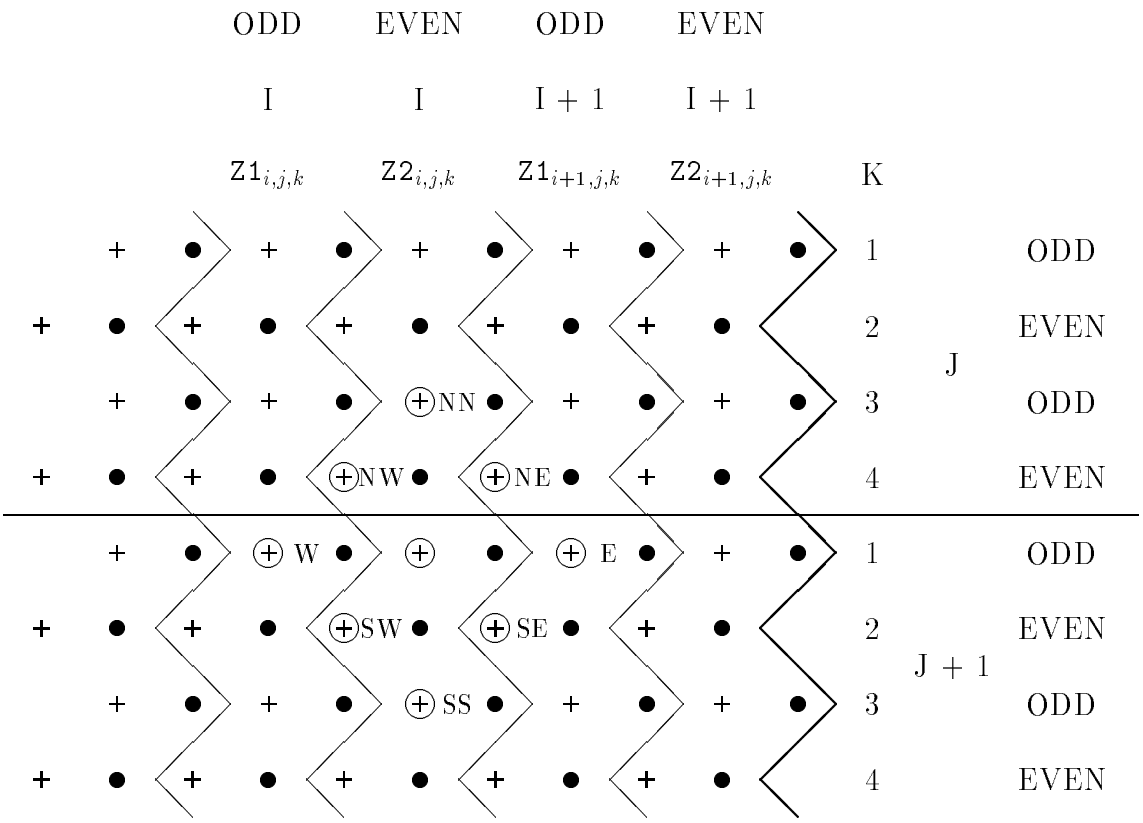


Figure 7 **Indexing on the 8-colour-grid**

Examples:

ζ lies on an array $\boldsymbol{Z2(I, J+1, 1)}$

ζ_{NN} lies on an array $\boldsymbol{Z2(I, J, 3)}$

ζ_{SE} lies on an array $\boldsymbol{Z2(I+1, J+1, 2)}$

The following steps 1 to 9 correspond to one iteration cycle.
Periodic boundary conditions require an update of one boundary after each step in the iteration procedure. $IH = (IE + 1)/2, \quad JM = (JE + 1)/2$.

1. $Z1(I, J, 1)$ is calculated starting from $I = 2$ to $I = IH$ and $J = 1$ to JM

- $Z1(1, J, 1)$ is updated from its eastern boundary

2. $Z2(I, J, 3)$ is calculated from $I = 1$ to $I = IH - 1$ and from $J = 1$ to JM

- $Z2(IH, J, 3)$ is updated from its western boundary

- Half sweep ODD-grid finished, adjust ω_{ODD}

3. $Z1(I, J, 3)$ is calculated starting from $I = 2$ to $I = IH$ and $J = 1$ to JM

- $Z1(1, J, 3)$ is updated from its eastern boundary

4. $Z2(I, J, 1)$ is calculated from $I = 1$ to $I = IH - 1$ and from $J = 1$ to JM

- $Z2(IH, J, 1)$ is updated from its western boundary

- Full sweep ODD-grid finished, adjust ω_{ODD}

5. $Z1(I, J, 2)$ is calculated starting from $I = 2$ to $I = IH$ and $J = 1$ to JM

- $Z1(1, J, 2)$ is updated from its eastern boundary

6. $Z2(I, J, 4)$ is calculated from $I = 1$ to $I = IH - 1$ and from $J = 1$ to JM

- $Z2(IH, J, 4)$ is updated from its western boundary

- Half sweep EVEN-grid finished, adjust ω_{EVEN}

7. $Z1(I, J, 4)$ is calculated starting from $I = 2$ to $I = IH$ and $J = 1$ to JM

- $Z1(1, J, 4)$ is updated from its eastern boundary

8. $Z2(Id, J, 2)$ is calculated from $I = 1$ to $I = IH - 1$ and from $J = 1$ to JM

- *Z2(IH,J,2)* is updated from its western boundary

- Full sweep EVEN-grid finished, adjust ω_{EVEN}

9. Go back to step 1. or stop if your convergence criterion matches your precision requirements (see below)

A full iteration cycle is indicated in Figure 8. Numbers 1-8 refer to iteration steps.

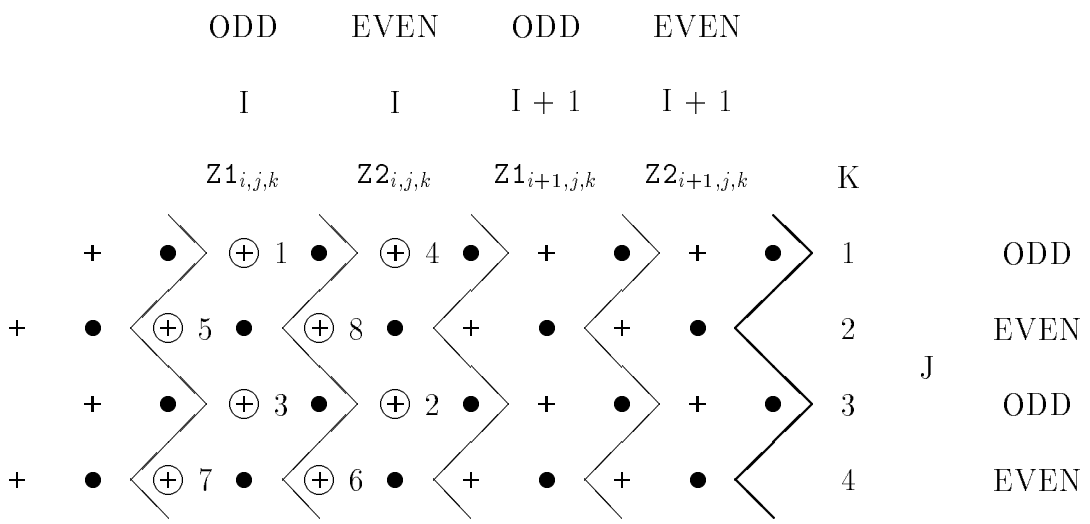


Figure 8 **Schematic ordering of a full iteration cycle**

6.4.2 Convergence criteria

Several, more or less equivalent criteria for iteration convergence can be defined (see e.g. Roache 1972, p. 174).

The convergence is monitored in HOPE by calculating the sum of the squares of the differences ("Euclidean norm") of two consecutive iteration cycles l and $l + 1$

$$C^{l+1} = \sum_{i,j} (\zeta_{i,j}^{l+1} - \zeta_{i,j}^{l})^2 \qquad (6.4.2.1)$$

and the total contraction $C^{END}/(MAX(C^l))$, which should be small. END is the number of iterations found to be necessary during testing and should be several times the largest horizontal dimension.

6.4.3 Getting started with the iterative solution

Run HOPE for one time step with *IELIMI = 0* and a number of different values for ρ_{Jacobi} (Variable *ROCON*). Reset the initial-guess field before iteration for each different value of *ROCON*. Monitor *C* and $C^{END}/MAX(C^l)$. The spectral radius ρ_{Jacobi} has to be adjusted with a precision of about 10^{-4}. For large *IE* or *JE* this value lies in the vicinity of 1 (typically near 0.995).

7 Stability of linear free waves

In this section, we study the propagation characteristics of a wave like disturbance $exp(ikx + ily)$ on an f-plane. The discretisation replaces ∂_x by $(2i\sin(k\Delta x/2))/\Delta x$ and ∂_y by $(2i\sin(l\Delta y/2))/\Delta y$. With the abbreviations $F = f\Delta t$, $X = 2\sin(k\Delta x/2)(\Delta t/\Delta x)$ and $Y = 2\sin(l\Delta y/2)(\Delta t/\Delta y)$, the system reads

$$AW^{n+1} = BW^n \tag{7.1}$$

where

$$W = \begin{pmatrix} U \\ V \\ P \end{pmatrix} \tag{7.2}$$

$$A = \begin{bmatrix} 1 & -\alpha F & i\alpha gX \\ \alpha F & 1 & i\alpha gY \\ i\beta HX & i\beta HY & 1 \end{bmatrix} \tag{7.3}$$

$$B = \begin{bmatrix} 1 & (1-\alpha)F & -i(1-\alpha)gX \\ -(1-\alpha)F & 1 & -i(1-\alpha)gY \\ -i(1-\beta)HX & -i(1-\beta)HY & 1 \end{bmatrix} \tag{7.4}$$

The stability of the scheme is given by the eigenvalues λ_i, $i = 1,2,3$ of the time-stepping matrix $C = A^{-1}B$. A full eigenanalysis would require to solve for the roots of the full characteristic polynomial. The stability properties can be found, however, by a simplified consideration:
It is immediately seen that the geostrophic mode remains unchanged, i.e., one of the three eigenvalues, say λ_3 is 1. The product of all eigenvalues is

$$\lambda_1\lambda_2\lambda_3 = det(C) = det(B)/det(A) = \frac{1 + (1-\alpha)^2F^2 + (1-\alpha)(1-\beta)gH(X^2+Y^2)}{1 + \alpha^2F^2 + \alpha\beta gH(X^2+Y^2)}$$

For $\alpha, \beta > 0.5$, the product is less than unity. The sum of eigenvalues is given by the sum of the diagonal entries:

$$S = \lambda_1 + \lambda_2 + \lambda_3 = C_{11} + C_{22} + C_{33} \tag{7.5}$$

$$A^{-1} = det(A)^{-1}\begin{bmatrix} 1+\alpha\beta gHY^2 & \alpha(-\beta gHXY+F) & -\alpha(i\alpha gY+igX) \\ \alpha(-\beta gHXY-F) & 1+\alpha\beta gHX^2 & \alpha(i\alpha gX+igY) \\ \beta(i\alpha HYF-iHX) & -\beta(i\alpha FHX+iHY) & 1+\alpha^2F^2 \end{bmatrix} \tag{7.6}$$

$$S = det(A)^{-1}(3-2F^2\alpha(1-\alpha)+\alpha^2F^2-gH(X^2+Y^2)(\alpha(1-\beta)+(1-\alpha)\beta-\alpha\beta)) \tag{7.7}$$

Subtracting λ_3 we get

$$S_2 = \lambda_1+\lambda_2 = \frac{2-2F^2\alpha(1-\alpha)-gH(X^2+Y^2)(\alpha+\beta-2\alpha\beta)}{1+\alpha^2F^2+\alpha\beta gH(X^2+Y^2)} \tag{7.8}$$

Inserting $\lambda_1 = det(C)\lambda_2^{-1}$, we get

$$\lambda_2^2 - S_2\lambda_2 + det(C) = 0 \tag{7.9}$$

which yields

$$\lambda_2 = \frac{S_2}{2} \pm \sqrt{\frac{S_2^2}{4} - det(C)} \tag{7.10}$$

With $\alpha, \beta \geq 0.5$ we have $S_2 \leq 2$, but $det(C) \geq |S_2/2|$. λ_1 and λ_2 are thus conjugated complex numbers with modulus less or equal unity.

The optimal choice of the parameters for neutral wave propagation is $\alpha = \beta = 0.5$, for which the numerical system is neutrally stable. Starting the model from a given “observed” density field creates normally a situation where mass field and topography are not in dynamical balance, which leads to intense adjustment currents. A parameter choice of $\alpha = \beta = 1$, which damps the system, might therefore be of help during any initial adjustment. Note however, that the stability of the scheme does not guarantee a correct phase propagation of waves: with $\alpha = \beta = 0.5$ the system may develop a computational mode similar to that of the leap frog scheme. It is therefore recommended to set both parameters greater than 0.5.

8 Sea-ice model

HOPE contains a sea-ice model (subroutine OCICE) which is called in subroutine OCTHER if *ICESEA = 1*.

OCICE solves the sea-ice momentum equation and the continuity equations for ice thickness, compactness, and snow depth.

OCICE also calls subroutine GROWTH, where the atmospheric freshwater and heat flux into the ocean, as well as thermodynamic changes of ice thickness and compactness, and of snow depth are calculated.

8.1 Sea-ice dynamics

Following Hibler (1979), the ice dynamics are formulated with the assumption that the ice cover behaves like a non-linear viscous-plastic compressible fluid with an elliptical yield curve and a no-stress condition for pure divergence. For further details the reader is referred to Hibler (1979) and references therein. The spatial discretisation of the dynamical equations described in the following is on the Arakawa-E grid of the ocean model.

8.1.1 Ice momentum equation

In the sea-ice momentum equation

$$\frac{\partial}{\partial t}\vec{v}_I = -f\vec{k}\times\vec{v}_I + \frac{\vec{\tau}_a}{\rho_I h_I'} + \frac{\vec{\tau}_o}{\rho_I h_I'} + \frac{\vec{F}_I}{\rho_I h_I'} - g\cdot\vec{\nabla}_H\zeta, \qquad (8.1.1.1)$$

the advection terms are neglected. $\vec{v}_I$ is the two-dimensional ice velocity, $\vec{\tau}_a$ the stress at the atmosphere-ice interface, and $\vec{\tau}_o = -\vec{\tau}_I$ (see section 4.2) the ice-water stress. $\vec{F}_I$ is the force due to variations of internal ice stress. ρ_I and h_I' are the ice density and the ice thickness.

As in Hibler (1979), the internal ice stress is modelled in analogy to a non-linear viscous compressible fluid obeying the constitutive law

$$\sigma_{ij} = 2\eta_I\dot{\varepsilon}_{ij} + (\zeta_I - \eta_I)\dot{\varepsilon}_{kk}\delta_{ij} - P_I\delta_{ij}/2\,, \qquad (8.1.1.2)$$

where σ_{ij} is the two-dimensional stress tensor, $P_I/2$ an ice-pressure term, and ζ_I and η_I non-linear bulk and shear viscosities.

$$\dot{\varepsilon}_{ij} = \begin{bmatrix} u_{I_x} & \dfrac{(u_{I_y} + v_{I_x})}{2} \\ \dfrac{(u_{I_y} + v_{I_x})}{2} & v_{I_y} \end{bmatrix}$$

is the symmetric strain rate tensor and $\dot{\varepsilon}_{kk}$ its trace.

Using this constitutive law, the force components due to the internal ice stress $F_I^i = \partial\sigma_{ij}/\partial x_j$ are

$$F_I^x = \frac{\partial}{\partial x}\left[(\eta_I + \zeta_I)\frac{\partial u_I}{\partial x} + (\zeta_I - \eta_I)\frac{\partial v_I}{\partial y} - \frac{P_I}{2}\right] + \frac{\partial}{\partial y}\left[\eta_I\left(\frac{\partial v_I}{\partial x} + \frac{\partial u_I}{\partial y}\right)\right]. \tag{8.1.1.3}$$

$$F_I^y = \frac{\partial}{\partial y}\left[(\eta_I + \zeta_I)\frac{\partial v_I}{\partial y} + (\zeta_I - \eta_I)\frac{\partial u_I}{\partial x} - \frac{P_I}{2}\right] + \frac{\partial}{\partial x}\left[\eta_I\left(\frac{\partial u_I}{\partial y} + \frac{\partial v_I}{\partial x}\right)\right]. \tag{8.1.1.4}$$

Following Hibler (1979), we define

$$\zeta_I = P_I/2\Delta_I, \tag{8.1.1.5}$$

$$\eta_I = \zeta_I/e^2, \tag{8.1.1.6}$$

$$\text{with} \qquad \Delta_I = [(\dot{\varepsilon}_{11}^2 + \dot{\varepsilon}_{22}^2)(1 + e^{-2}) + 4\dot{\varepsilon}_{12}^2/e^2 + 2\dot{\varepsilon}_{11}\dot{\varepsilon}_{22}(1 - e^{-2})]^{1/2} \quad . \tag{8.1.1.7}$$

This defines an elliptical yield curve with length ratio *e* of the principal axes.

The ice strength is coupled to the state of the ice cover by specifying the ice pressure to be a function of compactness and thickness:

$$P_I = P^* h_I' \cdot e^{-\tilde{C}(1 - A_I)}, \tag{8.1.1.8}$$

where P^* and $\tilde{C}$ are empirically derived constants.

The time discretisation of the momentum equations is implicit because of the large viscosities. The equations are solved by functional iteration with underrelaxation. At the North Pole (see section 9.1.2), a free-slip condition is specified.

8.1.2 Continuity equations

Continuity equations are solved for the mean ice thickness $h_I = h_I' \cdot A_I$, compactness A_I, and snow depths h_S:

$$\frac{\partial h_I}{\partial t} = -\nabla(\vec{v}_I h_I)\,, \tag{8.1.2.1}$$

$$\frac{\partial A_I}{\partial t} = -\nabla(\vec{v}_I A_I)\,, \tag{8.1.2.2}$$

$$\frac{\partial h_S}{\partial t} = -\nabla(\vec{v}_I h_S)\,. \tag{8.1.2.3}$$

These equations are also discretized implicitly and solved by functional iteration, e.g.

$$h_I^{n+\frac{l+1}{m}} = h_I^{n+\frac{l}{m}} - \Delta t \nabla\left(\vec{v}_I^{\,n+1} h_I^{n+\frac{l+1}{m}}\right). \tag{8.1.2.4}$$

For the horizontal discretisation, an upstream scheme is used, thus no explicit diffusion is required for stability reasons. In order to avoid excessive ice accumulation in narrow bays or semi-enclosed coastal points, however, it can be useful to add some explicit diffusion. This is done by setting $DHICE \neq 0$. There will be no diffusive flux across the ice edge.
Thermodynamic changes of ice thickness, compactness, and snow depth are computed in the thermodynamic part of the ice model (subroutine GROWTH).

8.2 Sea-ice thermodynamics

In case the ice model is activated (*ICESEA=1*), the changes of upper ocean potential temperature and salinity, and surface elevation which are due to atmospheric heat and freshwater fluxes are calculated in subroutine GROWTH, which is called in subroutine OCICE.
The change of surface elevation due to atmospheric freshwater fluxes is not modified by snow fall or melt nor by ice growth or melt. Accordingly, snow and ice are not explicitly accounted for in the pressure terms of the ocean dynamic equations.
Snow and ice albedos can take two values each, one for surface melting conditions ($T_S = T_{melt}$) and one for cold ($T_S < T_{melt}$) skin temperature T_S.
The formulation of the thermodynamic ice growth and the change of snow depth is based on that of Owens and Lemke (1990).

8.2.1 Snow cover

If a snow layer is included (*ISNFLG=1*), the precipitation is transformed into snow fall on the ice-covered part of each grid cell, in case the atmospheric near surface temperature is below T_{melt}. At surface melting conditions, i.e. if the diagnosed skin temperature is above T_{melt}, the surface heat-balance residual heat flux first melts the snow. If all ice can be melted in a grid cell from below, the heat is used first to melt the snow.

Snow draft is converted into ice. In cases of net snow accumulation, and if the ocean can not provide enough heat to counteract the related ice growth, the ice might become too thick. In order to avoid a model blow-up, the snow draft is directly converted into a freshwater flux if the ice is already thicker than a specified value (parameter *HSNTOICE*). This must be smaller than the thickness of the first ocean layer.

8.2.2 Thermodynamic ice growth

In the present model version, heat fluxes inside and outside an 'ice region' are treated differently. The 'ice region' has to be defined appropriately, regarding the forcing data. Presently, the forcing data are derived from a stand-alone atmosphere model, which used 100% ice concentration within the observed ice edge at its lower boundary. The 'ice region' should therefore include at least the region which was defined to be ice-covered for the atmosphere. Inside the 'ice region', atmospheric heat fluxes are calculated from bulk formulae using the 2m air and dew-point temperatures, cloudiness, wind speed, and atmospheric pressure. The parameterisations are the same as those mentioned in Stoessel (1991), with separate formulations for water and snow or ice surfaces.

For the calculation of the conductive heat flow, the different conductivities of snow, κ_S, and ice, κ_I, are accounted for by using an effective ice thickness $h_{eff} = (h_I + h_S \cdot \kappa_I / \kappa_S) / A_I$.

The thermodynamic ice growth is calculated for n thickness categories with thickness $h_{eff}^k = h_{eff} \cdot (2k-1)/n, k=1,n$. The number of thickness categories can be specified by setting *ICELEV* in the compile script.

The vertical resolution of each thickness category is that of the so-called zero-layer model without thermal heat capacity described by Semtner (1976). It is assumed, that the vertical snow and ice temperature structure adjusts to the external forcing in one time step. The conductive heat flux Q_c through the ice/snow layer is thus $Q_c = \kappa_I (T_{freez} - T_S) / h_{eff}^k$ for each thickness category. T_{freez} is the freezing-point temperature of sea water which is taken to be constant, independent from salinity.
All heat fluxes are positive upwards in what follows.

Outside the ice region, the surface heat flux is the sum of a net atmospheric heat flux Q_o and a heat flux correction Q_Δ

$$Q_w = Q_o + Q_\Delta , \tag{8.2.2.1}$$

which are taken directly from the atmosphere model data.

Inside the ice region, on the ice free part of the grid cell, the surface heat flux Q_w is calculated using a heat balance equation

$$Q_w = (1-\alpha_w)Q_s + Q_l + \varepsilon_w \sigma \theta_1^4 + Q_{lat} + Q_{se} + Q_\Delta . \tag{8.2.2.2}$$

θ_1 is the upper ocean temperature. Q_s and Q_l are the incident shortwave and longwave radiation heat fluxes, Q_{lat} and Q_{se} are the latent and sensible turbulent heat fluxes over the water. α_w and ε_w are the albedo and thermal emissivity of the sea surface.

Inside the ice region, on the ice covered part of the grid cell, a similar heat budget including the conductive heat flux through the snow/ice layer is used to calculate the new snow or ice skin temperature:

$$0 = (1-\alpha_S)Q_s + Q_l + \varepsilon_S \sigma T_S^4 + Q_{lat} + Q_{se} - Q_c . \tag{8.2.2.3}$$

Q_s and Q_l are again the incident shortwave and longwave radiative heat fluxes, Q_{lat} and Q_{se} are the latent and sensible turbulent heat fluxes over the snow or ice surface. α_S and ε_S are the albedo and thermal emissivity of the snow or ice surface.

In case of surface melt, i.e. if $T_S > T_{melt}$, the heat budget equation is recalculated with the skin temperature set to T_{melt}:

$$Q_{res} = (1-\alpha_S)Q_s + Q_l + \varepsilon_S \sigma T_{melt}^4 + Q_{lat} + Q_{se} - Q_c . \tag{8.2.2.4}$$

Q_{res} is first used for surface melt of the snow layer and than of the ice layer. If all snow and ice can be melted from above, the remaining heat is added to Q_w and is used to update the oceanic temperature.

The conductive heat flux Q_c and the atmospheric heat flux through the air/water interface, Q_w, are used to update the upper ocean temperature:

$$\hat{\theta}_1 = \theta_1^{n+1/m} - \frac{\Delta t (Q_w + Q_c)}{\rho_o c_p (d_{w1} + \zeta - h_{draft}^{n+1/m})} - \frac{h_I^n \rho_I L_f}{\rho_o c_p (d_{w1} + \zeta - h_{draft}^{n+1/m})} \tag{8.2.2.5}$$

L_f is the heat of fusion of sea ice and h_{draft} is the snow/ice draft. n indicates the old time level.

The ocean temperature can not drop below freezing point, thus in case $\hat{\theta}_1 < T_{freez}$:

$$\theta_1^{n+(l+1)/m} = T_{freez} \tag{8.2.2.6}$$

and the residual heat content in the upper ocean layer is used to calculate the new area-mean ice thickness:

$$h_I^{n+(l+1)/m} = \frac{(T_{freez} - \hat{\theta}_1)}{\rho_I L_f} \cdot \rho_o c_p (d_{w1} + \zeta - h_{draft}^{n+l/m}) \tag{8.2.2.7}$$

If $\hat{\theta}_1$ is above the freezing-point, all snow and ice is melted before the new ocean temperature can raise above T_{freez}:

$$\theta_1^{n+(l+1)/m} = \hat{\theta}_1 - Min\left(\frac{h_I^{n+l/m}\rho_I L_f}{\rho_o c_p (d_{w1} + \zeta - h_{draft}^{n+l/m})}, \hat{\theta}_1 - T_{freez}\right) \tag{8.2.2.8}$$

and the new ice thickness is

$$h_I^{n+(l+1)/m} = Max\left(h_I^{n+l/m} - (\hat{\theta}_1 - T_{freez})\frac{\rho_o c_p (d_{w1} + \zeta - h_{draft}^{n+l/m})}{\rho_I L_f}, 0\right). \tag{8.2.2.9}$$

8.2.3 Thermodynamic lead opening/closing

The thermodynamic lead opening and closing is similar to that described by Hibler (1979).
In grid cells containing ice, the thermodynamic compactness changes are calculated from the freezing rate over water, and from the thick-ice melting rate.

The rate of lead closing is

$$S_{A_I}^{thin} = Max\left(\frac{\Delta h_I^{thin}(1 - A_I^n)}{h_o \Delta t}, 0\right) \tag{8.2.3.1}$$

where $\Delta h_I^{thin} = \Delta t Q_w / (\rho_I \cdot L_f)$. Thus, thermodynamic lead-closing occurs for freezing conditions over open water only. h_o is the minimum thickness of newly created ice.

The rate of lead opening is (with $\Delta h_I^{thick} = h_I^{n+1} - h_I^n$)

$$S_{A_I}^{thick} = Min\left(\frac{\Delta h_I^{thick} A_I^n}{2 h_I^n \Delta t}, 0\right). \tag{8.2.3.2}$$

Thus, thermodynamic lead opening occurs only if thick ice is melted. It is assumed here that the ice thickness distribution varies linearly between *0* and $2h_I$, and that the melting rate is independent on the ice thickness.

The compactness A_I is updated by the sum of both term:

$$\frac{\partial A_I}{\partial t} = S_{A_I}^{thin} + S_{A_I}^{thick}. \tag{8.2.3.3}$$

8.3 Update of salinity

The salinity of sea ice, $S_{ice,}$ is assumed to be independent of its age and of the salinity of the water from which it is formed. Since $S_{ice} << S_1$ usually, melting of sea ice is accompanied by a decrease of salinity of the surrounding sea water, while freezing results in an increase of salinity.

The salinity changes due to haline rejection and precipitation (modified by snow fall/melt) are calculated in subroutine GROWTH as

$$S_1^{n+(l+1)/m} = S_1^{n+l/m} + \Delta S \tag{8.3.1}$$

with ΔS obeying

$$\begin{aligned}(S_1^{n+l/m} + \Delta S) \cdot (d_{w1} + \zeta^{n+1/m} - h_{draft}^{n+(l+1)/m}) + \frac{\rho_I \cdot h_I^{n+(l+1)/m}}{\rho_w} S_{ice} \\ = S_1^{n+l/m}(d_{w1} + \zeta^n - h_{draft}^{n+l/m}) + \frac{\rho_I \cdot h_I^{n+l/m}}{\rho_w} S_{ice}\end{aligned} \tag{8.3.2}$$

where $\zeta^{n+1/m}$ is the surface elevation at the old time level updated by the atmospheric freshwater flux. In subroutine GROWTH, the brine released during freezing of sea ice is injected into the uppermost layer only. In order to improve the characteristics of the water masses formed in regions with ice production, it can be useful to transfer some of the brine into deeper layers. This can be done by calling subroutine OCBRINE (with *IOCBRINE=1)* and specifying the distribution profile in subroutine OCEINI. (array WEI). If the profile is to depend on the actual state of the ocean, it must be calculated in subroutine OCBRINE.

8.4 Subroutines

In this section, the subroutines are listed which are called in connection with the ice model.

8.4.1 Subroutine BUDGET

BUDGET calculates the snow or ice skin temperature and the surface residual heat flux at the snow/ atmosphere or ice/atmosphere interface, using a surface heat balance equation (see equations (8.2.2.3) and (8.2.2.4)). From this, the surface melt of snow or ice and the conductive heat flux for bottom ablation or accretion of sea ice is deduced.

8.4.2 Subroutine GROWTH

Update of sea-surface elevation by atmospheric freshwater fluxes

Update of upper ocean layer temperature due to atmospheric heat fluxes and ice growth or melt

Update of upper ocean salinity due to atmospheric freshwater fluxes, snow melt, and ice growth or melt

Update of snow thickness due to snow fall or melt

Update of ice thickness due to thermodynamic change of thick ice and new-ice growth

Update of ice concentration due to lead opening and closing

Update of ice and snow thickness due to snow-to-ice conversion

8.4.3 Subroutine ICPODI

Some additional smoothing of the ice variables near the North Pole is performed in this subroutine.

8.4.4 Subroutine OBUDGET

OBUDGET calculates the atmospheric heat flux through the water/atmosphere interface by either using net atmospheric heat flux data or a surface heat flux balance. The output is in units of ice growth (m/s).

8.4.5 Subroutine OCBRINE

If *IOCBRINE = 1*, the brine 'released' during freezing of sea ice is transferred into deeper layers. The distribution profile can be specified in subroutine OCEINI in case it is independent of the state of the ocean.

8.4.6 Subroutine VAPOR

Depending on the parameters passed to subroutine VAPOR, the water vapor pressure in the atmosphere or the saturation vapor pressure over the water or the ice are calculated.

9 User's manual

The ocean model is advanced by 1 time step in subroutine OCESTEP. OCESTEP is called by the control program OCEMAST which also calls the initialisation subroutine OCEINI.
In OCEINI all parameters and options are set and the NAMELIST variables are read in. OCEINI also calls subroutine AUFR to read a restart file or it reads the initial stratification and specifies the initial ice cover in the case no restart file will be read (*IAUFR = 0*). In addition, OCEINI reads annual mean SSS data which are used for model SSS relaxation to observed data in subroutine OCTHER and it initializes output time-series arrays and arrays for output-field accumulation.

There are 4 additional subroutines which are called by OCEINI. These are

subroutine BELEG
subroutine BODEN
subroutine CORIOL,

and, depending on whether the barotropic system is solved directly or iteratively,

subroutine TRIAN or
subroutine OCITPRE.

The purpose of these subroutines will be explained in the following subsections.

The forcing data (except climatological data) are read by a pseudo-atmosphere. The initialisation of the pseudo-atmosphere (subroutine ATMINI) is also done in OCEMAST, as is that of the interface which interpolates the data from the atmosphere to the ocean grid and vice versa (subroutine COUPINI). The integration is done in a loop whose length (’number of coupled time steps’) must be specified in the interfaces NAMELIST. Fields are exchanged once each ’coupled’ time step. At the end of the integration loop, OCEMAST calls the integration-postprocessing subroutines which close the output data sets.

The length of the standard-output file can be controlled by the variables *NWFREQ* and *NDRUCK*. *NWFREQ* is the printing frequency of mean, extreme, instantaneous, and integrated variables. *NDRUCK* is the frequency of 2D-field printing. By setting the arrays *NPR**, subgrids for 2D-field printing can be specified.

In the following, we describe the steps required to run the model with arbitrary geometry:

1. Specify the grid (3-D)
2. Provide the input data sets
3. Specify the reference stratification

4. Set the switches
5. Choose the stability parameters
6. Choose the mixing coefficients
7. Set the number of timesteps

9.1 Grid specification

The following dimensions have to be specified in the include file PARAM1.h:

IE: number of grid points in west-east direction in each subgrid (EVEN/ODD)
JE: number of grid points in north-south direction in each subgrid (EVEN/ODD)
KE: number of layers
KBB: bandwidth of barotropic system matrix
ILL:total number of wet scalar grid points

The bandwidth of the barotropic system matrix and the total number of wet scalar grid points are calculated in subroutine BODEN. If the initially specified dimensions are too small the model will STOP there.

Layer thicknesses Δz_w (*DZW*) have to be specified in OCEINI. The reference stratification *TAF* and *SAF* has to be specified on the resulting *TIESTU* levels (see section “Vertical discretisation” on page 10 for further details).

All input files have to be provided on the grid defined by $ALAT$ and $ALONG$ described in the following.

9.1.1 West-east grid configuration

The model needs two extra grid points in each subgrid in west-east direction to represent solid or cyclic boundaries. These points are $I = 1$ and $I = IE$. For cyclic boundaries the location of grid points $I = 2$ is identical to that of $I = IE$ and the location of point $I = 1$ is identical to that of $I = IE - 1$. If the model uses cyclic boundary conditions in west-east direction, the parameter $ICYCLI$ must be set to *1*. This activates calls to the subroutines OCPODI, OCDIP2, ICPODI, PERIO3, PERIO2, PERIO32, PERIO3P, and PERJTO. These subroutines smooth variables near the North Pole or copy values from interior points onto the extra boundary points. The longitudes on array *ALONG(I), I = 2,...,5* correspond to the locations of scalar and vector points of the first two grid rows in the E-grid configuration, i.e. the first odd vector point has the longitude of $ALONG(4)$. See section “Horizontal discretisation” on

page 7 and the Table on page 8 for a more detailed description of the E-grid and the relative positions of vector and scalar points.

If the model area has no cyclic boundaries, the western-most oceanic vector point (even, $I = 2$) has longitude $ALONG(5)$.

9.1.2 North-south grid configuration

Two extra rows are needed in this direction because the North Pole is treated currently as a solid wall. (A version of HOPE with a 'wet' North Pole is in preparation.) ALAT(4) should be 90^{o} N, which is the meridional location of the second even row. In the vicinity of the North Pole (*J = JGEOO,JGEOU* of each subgrid), the thermohaline variables are zonally averaged (for $NPOLE = 1$).

If the North Pole is not included in the model area, the subroutine OCPODI must not be called! Set $NPOLE = 0$.

9.1.3 Subroutine BELEG

The first input file to be created is DEPATL, which should contain a vector of monotonically decreasing latitudes $ALAT(2 \cdot JE)$, a vector of monotonically increasing longitudes $ALONG(2 \cdot IE + 6)$, and an array containing ocean depths (positive).
The bathymetry can be provided on the vector or scalar grid points. The depths at the other grid must than be calculated to be the maximum of the surrounding vector depths or the minimum of surrounding scalar depths. In both cases, the minimum and maximum operations must be inverse to each other.

If *HOPE* is not being used in its standard configuration, this file has to be provided by the user.

Meridional and zonal grid distances are calculated and stored on two-dimensional arrays.

Vertical levels for vector and scalar variables are computed from the given layer thicknesses $DZW(KE)$. Reference pressure and density are calculated from reference salinities and temperatures at depths *TIESTU*. The reference salinities *SAF* and temperatures *TAF* should be specified in subroutine OCEINI.

A couple of arrays are set to zero initially.

The following arrays are initialized with non-zero values

- $TURB_{E/O} = 10^{-12}\mathrm{m}^2\mathrm{s}^{-1}$
- $TH_{E/O} = TAF$
- $SA_{E/O} = SAF$.

TAF and *SAF* are used as initial stratification if no restart file and no 3D initial stratification is read by subroutine OCEINI.

9.1.4 Subroutine BODEN

The topography at scalar points is recalculated to be the maximum depth of the four surrounding vector points.
All scalar points are consecutively numbered and the number of rows *ILL* and the bandwidth *KBB* of the barotropic system matrix are calculated and printed. Modify PARAM1.h according to the numbers given (see section 6.3.2).
Grid cell heights at scalar ($DDP_{E/O}$) and vector points ($DDU_{E/O}$), land/sea masks at scalar ($WET_{E/O}$) and vector points ($AMSU_{E/O}$), and surface areas of scalar ($AREP_{E/O}$) and vector cells ($AREU_{E/O}$) are calculated.

For diagnostic purposes the total number of vector and scalar points in each layer is calculated.

A vertical profile is calculated which allows the penetration of solar radiation into sub-surface layers (for $ISOLPR = 1$).

$$\frac{I}{I_O} = (1 - R)e^{(z/D)} \qquad (9.1.4.1)$$

Each component of the vector $SOLPROF$ is initialized with the fraction of incident radiation which is absorbed in that level; the surface level is initialized with the negative value of the fraction of radiation that passes through it, since all incident solar radiation is put into the surface layer first.

However, a correction to the profile must be made in shallow waters. In such areas, some of the solar radiation will reach the ocean floor. This fraction is not allowed to disappear through the sea bed, but is added to the lowest ocean layer.

$SOLPEN$ is the number of levels through which penetration of solar radiation is allowed. $RPEN$ is the fraction of radiation that is absorbed near the surface; the remainder is assumed to decay exponentially, with a length scale $DPEN$ (Paulson and Simpson, 1977). $NSOLPEN$ should be large enough to allow penetration to depths where the solar flux is small compared to the diffusive flux of heat.

9.1.5 Subroutine CORIOL

This subroutine sets the Coriolis parameter on all grid points and initializes the arrays in which it is used.

9.1.6 Subroutine TRIAN

The entries of the system matrix are set here. The matrix is then triangularized (see “Barotropic subsystem” on page 33.). See section “Getting started with the direct solution” on page 40 on how to get started with the direct solution and how to run the model with the matrix in core memory or on disk.

9.1.7 Subroutine OCITPRE

Auxiliary arrays for the iterative solution of the system matrix are defined here.

9.2 Input data sets

Table 2

Unit	File	Description	Comment
5	ATMCTL	Pseudo-Atmosphere NAMELIST file	
9	OCEIN	Ocean NAMELIST file	
10	KONTCTL	Interface NAMELIST file	
11	ATMGRID	Atmosphere grid specification	used by subroutine COUPINI only
12	OZSCAL	Ocean scalar grid specification	"
13	OZVECT	Ocean vector grid specification	"
14	RESTARTa	Restart file	read/written by subroutines AUFR/AUFW
15	RESTARTb	Restart file	subroutine AUFW writes alternatively on unit 14 and 15
17	DEPATL	Longitudes, latitudes and topography [^{o}N/^{o}E/m]	
19	MATRIX	Elimination factors and triangularized barotropic system matrix	used only if the matrix is not core resident

Table 2

Unit	File	Description	Comment
21	INITEM	3-D initial temperature field [^{o}C]	used only at 1. run
22	INISAL	3-D initial salinity field [psu]	used only at 1. run
23	FWFCORR	Annual freshwater flux (e.g. flux correction field) [m/sec]	used if in addition to/ instead of P-E and salinity relaxation an annual mean freshwater flux is applied
25	PATEMP	Monthly surface temperature field [^{o}C]	optional for relaxation to climatological SST
26	SURSAL	Annual surface salinity field [psu]	used for freshwater flux relaxation
27	PAWIND	Monthly wind stress field [Pa]	x- and y- components
IFLXUNIT	FLUXES	Daily atmospheric forcing fields	generated by ECHAM4
IRNOUNIT	RUNOFF	Daily continental runoff	generated by ECHAM4

All of these files are not necessarily used in each run:
Files INITEM and INISAL are used only if IAUFR = 0, i.e. in the very first run of the model.
File FWFCORR is not used if IFWFCORR = 0.
File SURSAL is not used if IRELSAL = 0.
File MATRIX is not used if $IMATW \neq 0$.
Files PATEMP and PAWIND are not used if atmospheric forcing data are applied globally.

9.2.1 Initialisation of thermohaline fields

HOPE may be initialized with annual mean temperature and salinity fields stored in files INITEM and INISAL:

```
      READ (21) (((THE(I,J),I=2,IE-1),J=1,JE),K=1,KE),
     +          (((THO(I,J),I=2,IE-1),J=1,JE),K=1,KE)
      READ (22) (((SAE(I,J),I=2,IE-1),J=1,JE),K=1,KE),
     +          (((SAO(I,J),I=2,IE-1),J=1,JE),K=1,KE)
```

9.2.2 Temperature forcing

Temperature forcing data file PATEMP should contain monthly mean values of 12 consecutively written

unformatted records (January = record 1,..., December = record 12).

```
      READ (25) ,((TAFE(I,J),I=2,IE-1),J=1,JE),
     +           ((TAFO(I,J),I=2,IE-1),J=1,JE)
```

9.2.3 Salinity forcing

A freshwater flux is obtained by relaxing sea surface salinity to a prescribed climatology (e.g.Levitus, 1982/94) if $IRELSAL \neq 0$. Only annual mean values (1 record) can be provided in file SURSAL for the present version.

```
      READ (26) ((SAFE(I,J),I=2,IE-1),J=1,JE),
     +          ((SAFO(I,J),I=2,IE-1),J=1,JE)
```

9.2.4 Wind stress

Monthly mean values of wind stress (e.g. Hellerman and Rosenstein, 1983) on 12 consecutively written unformatted records (January = record 1,..., December = record 12) should be provided by file *PAWIND*.

```
      READ (27) ((TXE(I,J),I=2,IE-1),J=1,JE),
     +          ((TYE(I,J),I=2,IE-1),J=1,JE),
     +          ((TXO(I,J),I=2,IE-1),J=1,JE),
     +          ((TYO(I,J),I=2,IE-1),J=1,JE)
```

9.2.5 Forcing with atmospheric model data

HOPE can also be coupled to an atmosphere GCM, and a version of HOPE appropriate to be coupled to the ECHAM atmosphere model has been developed. In the version described here, where the ocean is driven by daily data produced by ECHAM, the forcing model data are read by a pseudo-atmosphere (subroutine ATMSTEP). Pseudo-atmosphere, ocean and the data exchange are controlled by program OCEMAST.

At each ’coupled’ time step

- all atmosphere forcing fields that will be passed to the ocean are set to 0 (subroutine SETNULL)
- the ocean fields used by the atmosphere are interpolated to the atmosphere grid (subroutine OATRANS)
- the atmospheric forcing fields for the new ’coupled’ time step are read (subroutine ATMSTEP)
- the atmospheric forcing fields are interpolated to the ocean grid (subroutine AOTRANS)
- the ocean fields that are passed to the atmosphere are set to 0 (subroutine OSSTINI); the ocean passes data averaged over one ’coupled’ time step, the accumulation is done in subroutine

OCESTEP by subroutine SSTACC

- the ocean model is integrated the number of ocean time steps that corresponds to one 'coupled' time step (*NTOCEAN* calls of subroutine OCESTEP).

The interpolation between the ocean and atmosphere model grids is based on an area-mean preserving quadratic scheme. This is supplemented by a linear extrapolation over non-matching sea points at land boundaries, and by a linear SST regression to distribute heat flux from the coarse resolution atmosphere grid onto the high resolution ocean grid.

9.3 Reference stratification

The following arrays have to be specified by the user with DATA-statements in subroutine OCEINI:

$TAF(KE)$: mean temperature at levels *TIESTU*.

$SAF(KE)$: mean salinity at levels *TIESTU*.

The reference stratification is used in the equation of state to determine the mean (reference) pressure at those depths. It can also be used as initial stratification.

9.4 Set switches

Switches are described in the section "User switches" on page 81.

The switch *IELIMI* defines the method of solution of the barotropic subsystem. It is recommended to read the sections "Direct solution of the barotropic system" on page 38 and "Iterative solution of the barotropic system" on page 41 and the subsections "Getting started with the direct solution" on page 40 and "Getting started with the iterative solution" on page 45 before changing this switch.

9.5 Stability parameters

Section "Stability of linear free waves" on page 47 may serve for a decision on how to choose the stability parameters α and β.

9.6 Mixing coefficients and bottom friction

Friction parameters depend on the grid size, time step and the physical processes to be studied. There is no recipe available for choosing vertical and horizontal mixing or bottom friction parameters. Recent lit-

erature on global or regional models may provide some guidance.

9.7 Time-step counter

Forcing data sets for climate studies with HOPE are currently provided as annual, monthly or daily mean fields.

The organization of the forcing fields in HOPE is based on a month-counting loop. For the monthly or annual data, it is assumed that a forcing field is stored on a file containing 12 (January = record 1,..., December = record 12) or 1 records, respectively (see section “Input data sets” on page 63).
The daily data are organized as monthly files with 30 records for each variable.

In the following, the counting of timesteps, months and years is described.

- *LYEAR* is the overall year of a (continued) experiment. If the ocean is initialized with climatological data, this is set to 0 before the start of the integration and is then incremented by 1 at the beginning of a new year. *LYEAR* is passed to/from restart files.
- *LMONTH* is the month of the current year. If the ocean is initialized with climatological data, this is set to 12 before the start of the integration and is then incremented by 1 (mod 12) at the beginning of a new month. *LMONTH* is passed to/from restart files
- *LDT* is the overall time step of a (continued) experiment. This is set to 0 before the start of a run; it is overwritten by the value read from the restart file and is incremented by 1 at each time step. *LDT* is passed to/from restart files.
- *NNNDT* is the time step of the current run. This is set to 0 before the start of the integration and is incremented by 1 at each time step. *NNNDT* is not passed to/from restart files.
- *NMONDT* is the time step of the current month. This is set to 0 before the start of the integration and is incremented by 1 (mod *NNN720*) at each time step. *NNN720* is the number of time steps per months and has to be specified in *PARAM1.h*. *NMONDT* is not passed to/from restart files.

9.8 Restart information

The restart file is read by subroutine AUFR if the switch $IAUFR$ is set to 1. Subroutine AUFR returns the last year *LYEAR* and month *LMONTH* which has been executed, and sets the pointer of the forcing data files to the actual month.

The monthly mean forcing data are linearly interpolated from month to month. This is done with the aid of auxiliary arrays (e.g. $TXE1$ and $TXE2$ for the zonal component of wind stress on even grid points).

In the interpolation scheme from monthly mean values to the actual time step, it is assumed that each data record represents data for the first time step of each month. This implies that the forcing data are shifted by half a month if data from the Levitus atlas or from Hellerman and Rosenstein are used.

All information required to restart the model is stored at the end of each month on a file by subroutine AUFW (for $IAUFW = 1$).Two files (switched each month) are available for this purpose to avoid loss of data in case of a system crash during writing on a restart file.

9.9 Output data sets

A time history file (ZEITexpid_YYYYM1M2) is generated during each run. expid is an experiment identifier which is defined in the job control script. YYYY is the current year and M1 and M2 are the first and last months of the present integration period. ZEITexpid_YYYYM1M2 contains variables at every time step .

Table 3

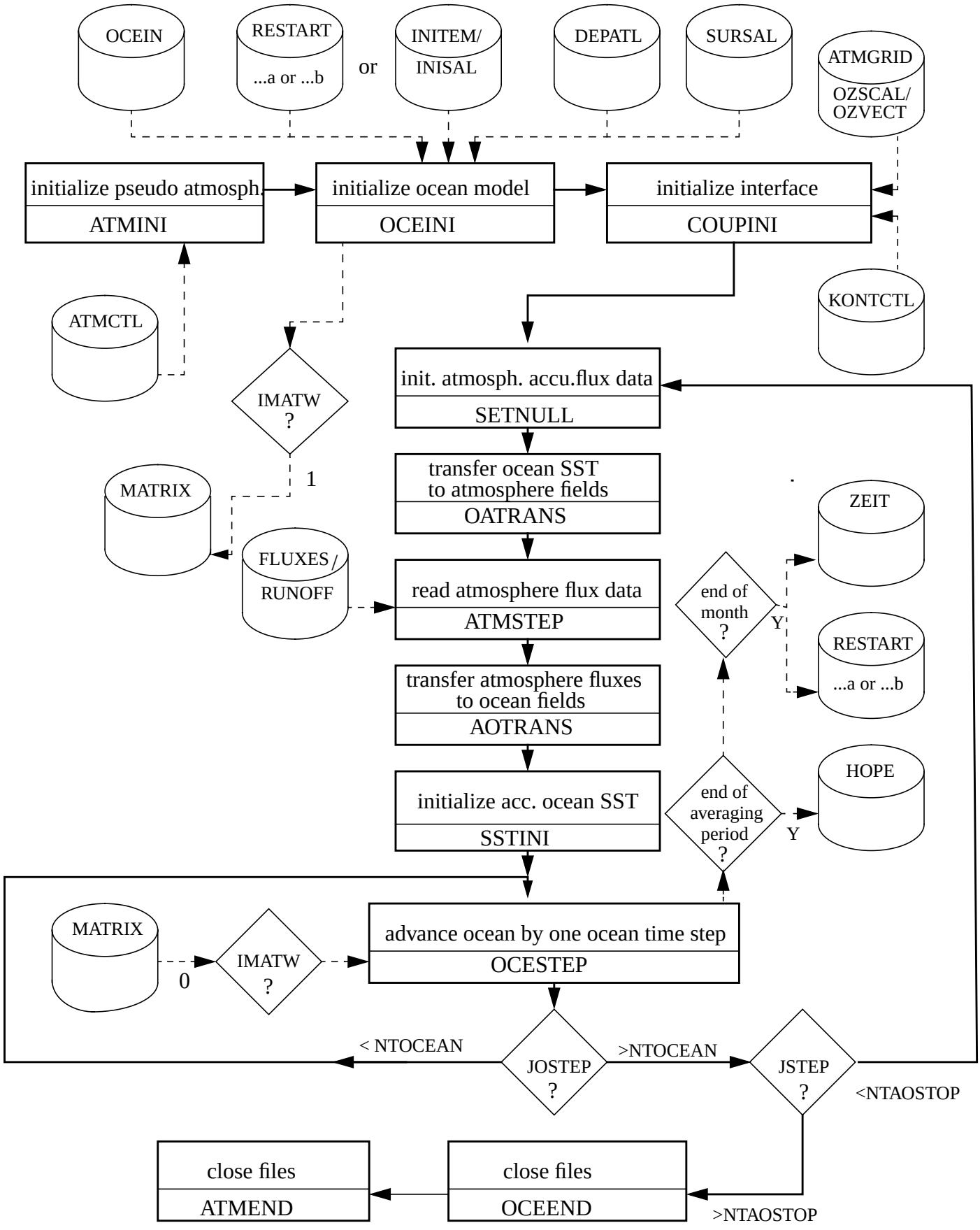
Unit	File	Comment
41	ZEITexpid_YYYYM1M2	Unformatted time series expid:Experiment identifier YYYY:Year M1:First month M2:Last month of the run
43	HOPEexpid_YYYYM1M2	GRIB-format output file of monthly mean 3d model fields expid:Experiment identifier YYYY:Year M1:First month M2:Last month of the run

9.10 Flow charts

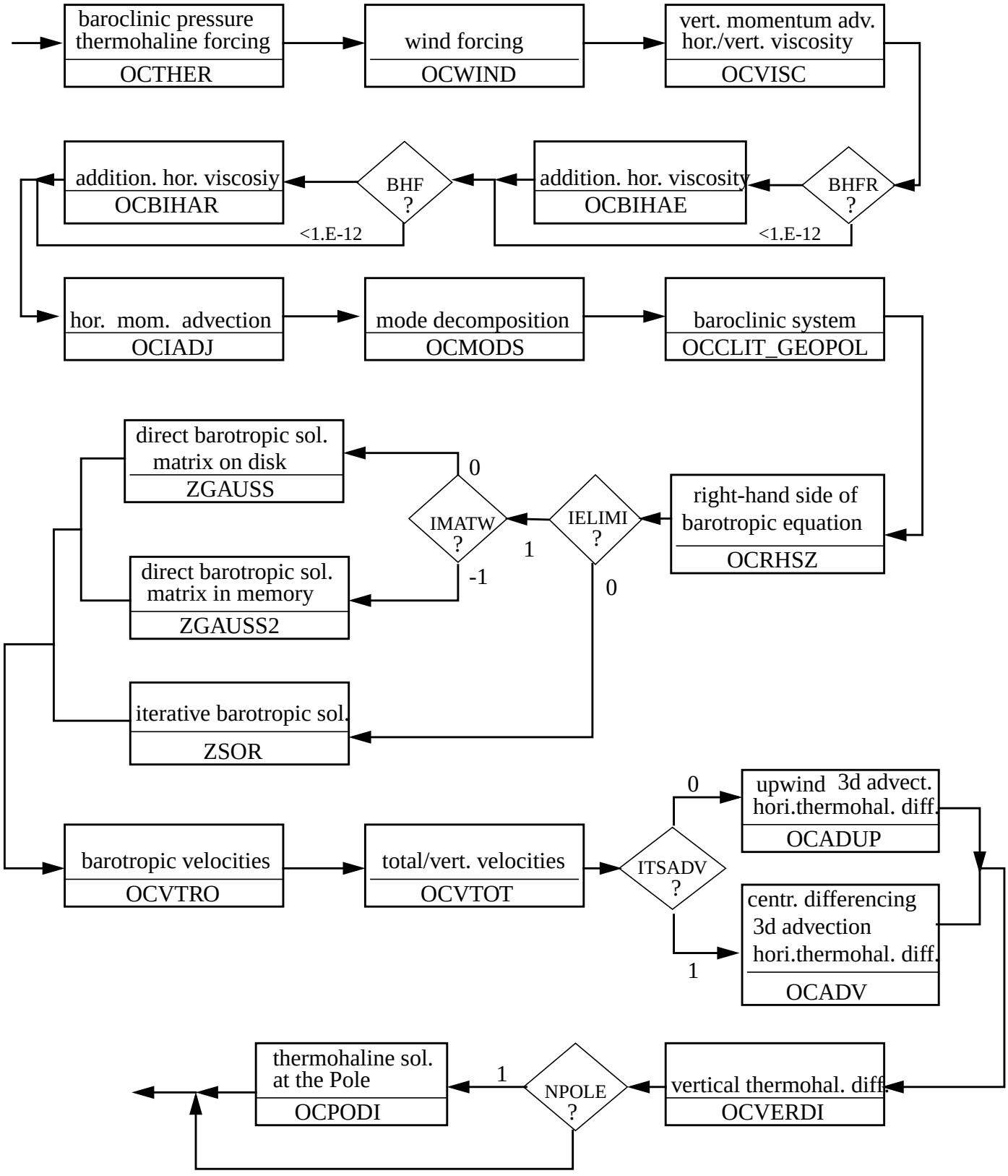
The flow chart of the main control program with a pseudo-atmosphere to read the forcing data and the disk I/O of the standard configuration is given in “Flow chart of the control program” on page 69. The subroutines involved are described in “Auxiliary subroutines for the ocean model” on page 73 and “Subroutines of the pseudo-atmosphere and the ocean-atmosphere interface” on page 78. The flow chart of the ocean time-stepping is given in “Flow chart of ocean time stepping” on page 70. The corresponding subroutines are described in “Time stepping” on page 13.

On the charts, solid arrows indicate the program flow, while dashed arrows indicate the disk I/O data-flow direction.

9.10.1 Flow chart of the control program



9.10.2 Flow chart of ocean time stepping



10 References

Abramowitz, M. and I. A. Stegun, 1972:
Handbook of Mathematical Functions,
Dover Publ. Inc., New York.

Arakawa, A. and V. R. Lamb, 1977:
Computational design of the basic dynamical processes of the UCLA general circulation model,
Meth. Comput. Phys., 17, 173-265.

Gill, A. E., 1982:
Atmosphere-Ocean Dynamics,
Academic Press, San Diego, California.

Hellerman, S. and M. Rosenstein, 1983:
Normal monthly wind stress over the world ocean with error estimates,
J. Phys. Oceanogr., 13, 1093-1104.

Hibler III, W. D., 1979:
A dynamic thermodynamic sea ice model,
J. Phys. Oceanogr., 9, 815-846.

Levitus, S., 1982:
Climatological Atlas of the World Ocean,
NOAA Professional Paper, 13, U.S. Department of Commerce.

Monin, A. S. and A. M. Yaglom, 1971:
Statistical fluid dynamics,
Vol. 1, The MIT Press.

Owens, W. B. and P. Lemke, 1990:
Sensitivity studies with a sea ice - mixed layer - pycnocline model in the Wedell Sea,
J. Geophys. Res., 95(C6), 9527-9538.

Pacanowski, R. C. and S. G. H. Philander, 1981:
Parameterization of vertical mixing in numerical models of tropical oceans,
J. Phys. Oceanogr., 11, 1443-1451.

Paulson, C. A. and J. J. Simpson, 1977:
Irradiance measurements in the upper ocean,
J. Phys. Oceanogr., 7, 952-956.

Press, W. H., B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, 1986:
Numerical recipes: The art of scientific computing,
Cambridge university press, New York.

Roache, P. J., 1972:
Computational Fluid Dynamics,
Hermosa Publ., Albuquerque, New Mexico.

Semtner, A., 1977:
A Model for the Thermodynamic Growth of Sea Ice in Numerical Investigations of Climate.
J. Phys. Oceanogr., 6, 379-389.

Sterl, A., 1991:
Manual for the primitive equation OGCM as used for ENSO studies and the interface to ECHAM, [Available from Max-Planck-Institut für Meteorologie, Bundesstraße 55, D-2000 Hamburg 13, Germany].

Stössel, A., 1991:
The Hamburg Sea-Ice Model,
Technical report, No. 3, German Climate Computer Center (DKRZ), Hamburg.

UNESCO, 1983:
Algorithms for computation of fundamental properties of sea water,
UNESCO Technical Papers in Marine Science, No. 44.

11Appendix A Auxiliary subroutines

11.1 Auxiliary subroutines for the ocean model

11.1.1 Subroutine ACCUMUL

Accumulation of output fields. The averaged fields will be saved on disk every *NWFREQ* time steps (subroutine GRIBOUT).

11.1.2 Subroutine ADISIT

Transformation of potential temperature θ to in-situ temperature T for use in the UNESCO equation of state.

11.1.3 Subroutine AUFR

Subroutine AUFR reads a restart-file from disk. The youngest file of RESTARTa and RESTARTb is read. Odd months read RESTARTa if the files contain the same data. Called if *IAUFR = 1*.

11.1.4 Subroutine AUFW

Subroutine AUFW writes a restart file on disk. If the last restart file read/written has been RESTARTa (unit 14), the next restart file will be written to RESTARTb (unit 15) and vice versa.
Called if *IAUFW = 1*.

11.1.5 Subroutine CFL

Check of the CFL-criterion in the zonal direction. If the criterion is grossly violated, the position of the largest CFL-parameter is found, and all model variables in a window centered on this position will be printed with the E-grid configuration.

11.1.6 Subroutine DRUCKD

Printing of model fields on the E-grid including a header. The size of the window to be printed, and the unit of the variable can be specified.

11.1.7 Subroutine DRUCKE

In subroutine DRUCKE, various model fields will be printed using subroutines DRUCKD or DRUCKF. DRUCKE is called every *NDRUCK* time steps.

11.1.8 Subroutine DRUCKF

Printing of model fields on a regular grid including a header. The size of the window to be printed, and the unit of the variable can be specified.

11.1.9 Subroutine OCDIPO

This subroutine is called by subroutine OCPODI and sets the values of the northernmost wet rows (*J=JGEOO,JGEOU*) of the 3d input field to their zonal mean value. This subroutine is used to make the zonal pressure gradient vanish at the North Pole. Called if *NPOLE = 1*.

11.1.10 Subroutine OCDIP2

Same as subroutine OCDIPO for 2d fields.

11.1.11 Subroutine OCEEND

Closure of the model output fields.

11.1.12 Subroutine OCEINI

Subroutine OCEINI initializes the ocean model:

- It sets parameter and switch defaults, and reads in the ocean NAMELIST OCECTL.
- It calculates the reference stratification
- It sets the standard layer thicknesses
- It calls the initialisation subroutines BELEG, BODEN, CORIOL
- It calls TRIAN if required
- It reads a restart file or the initial stratification
- It reads annual forcing fields, positions monthly forcing fields at the actual month, and reads the first month to be used
- It opens input and output files
- It specifies vertical profiles of viscosity and diffusivity coefficients *ABACKV* and *DBACKV*. These profiles are used as vertical eddy viscosity and diffusivity coefficients if the Richardson-no. dependent formulation is not used (*IEDDY = 0*). If *IEDDY = 1* is specified, they are used as background viscosity and diffusivity coefficients, respectively.

11.1.13 Subroutine OCESTEP

See “Time stepping” on page 13.

11.1.14 Subroutine OCTIMF

Internal storage management.

11.1.15 Subroutine OCTURB

Computation of the horizontal velocity shear for the shear-dependent horizontal eddy viscosity and diffusivity.

11.1.16 Subroutine OSSTACC

Accumulation of the ocean fields that will be passed to the atmosphere.

11.1.17 Subroutine OSSTINI

Initialisation of the ocean fields that will be passed to the atmosphere.

11.1.18 Subroutine OUTDIAG

Calculation and printing of diagnostics. Called every *NWFREQ* time steps.

11.1.19 Subroutine PERIO2

Update of cyclic boundaries of an array with dimension (IE, JE). See section “West-east grid configuration” on page 60 for details on the grid configuration. Called if *ICYCLI = 1*.

11.1.20 Subroutine PERIO3

Update of cyclic boundaries of an array with dimension (IE, JE, KE). See section “West-east grid configuration” on page 60 for details on the grid configuration. Called if *ICYCLI = 1*.

11.1.21 Subroutine PERIO3P

Update of cyclic boundaries of an array with dimension $(IE, JE, KE + 1)$. See section “West-east grid configuration” on page 60 for details on the west-east grid configuration. Called if *ICYCLI = 1*.

11.1.22 Subroutine PERIO32

Update of cyclic boundaries in one layer of an array with dimension (IE, JE, KE). See section “West-east grid configuration” on page 60 for details on the west-east grid configuration. Called if *ICYCLI = 1*.

11.1.23 Subroutine PERJTO

Update of cyclic boundaries of an array with dimension (*IE*, 2**JE*) . See section “West-east grid configuration” on page 60 for details on the grid configuration. Called if *ICYCLI = 1*.

11.1.24 Subroutine RHO1

Calculation of density with the equation of state defined by the Joint Panel on Oceanographic Tables and Standards (UNESCO (1981); see Gill (1982), appendix A3.1). Due to vectorisation requirements this subroutine calculates the density at all (EVEN or ODD) scalar points in one layer, even on land points. As a consequence, *T* and *S* should have sensible values on land points (e.g. *S* must be positive).

11.1.25 Function RHO

Calculation of density with the equation of state defined by the Joint Panel on Oceanographic Tables and Standards (UNESCO (1981); see Gill (1982), appendix A3.1). Function RHO calculates the density at one location only and is used to compute the reference stratification.

11.1.26 Subroutine VECMAX

The output array of VECMAX contains the maximum of the input vector and a specified constant.

11.1.27 Subroutine VECMAXC

Each entry of the output array of VECMAXC contains the maximum of the entries of the two input vectors.

11.1.28 Subroutine VECMIN

Same as *VECMAX*, but for the minimum.

11.1.29 Subroutine VECMINC

Same as VECMAXC, but for the minimum.

11.2 Subroutines for GRIB output

11.2.1 Subroutine GRIBEX

This subroutine ’degribs’ the GRIB-formatted forcing fields and performs an error check.

11.2.2 Subroutine GRIBOUT

Subroutine GRIBOUT prepares model output fields to be ready for conversion into GRIB format and writes them on disk.

11.2.3 Subroutine HEADER1

Subroutine HEADER1 sets GRIB-format header information on the experiment.

11.2.4 Subroutine HEADER2

Subroutine HEADER2 sets GRIB-format header information on the actual data field.

11.2.5 Subroutine PBGRIB

Subroutine PBGRIB reads the forcing data in GRIB format.

11.2.6 Subroutine PBOPEN

Subroutine PBOPEN opens GRIB-formatted input files.

11.2.7 Subroutine WRIGRIB

Subroutine WRIGRIB normalizes accumulated GRIB output fields and calls the subroutines HEADER2 and GRIBOUT to write header and data on disk. After writing, it resets the arrays to zero.

11.3 Subroutines of the pseudo-atmosphere and the ocean-atmosphere interface

The subroutines of the pseudo-atmosphere are described in detail in Sterl (1991).

11.3.1 Subroutine AGET

Transfer of forcing fields from atmosphere COMMON blocks to the interface.

11.3.2 Subroutine AOTRANS

Transfer of atmosphere accumulated fields to the ocean by a series of subroutine calls: AVERAGE, AGET, INTERPU, OPUT.

11.3.3 Subroutine APUT

Transfer of ocean SST to the atmosphere COMMON block ATMVAL.

11.3.4 Subroutine ATMEND

Closure of the atmosphere output files.

11.3.5 Subroutine ATMINI

Subroutine ATMINI sets the defaults of the atmosphere parameter and switches and reads the atmosphere NAMELIST ATMCTL.

11.3.6 Subroutine ATMREAD

Subroutine ATMREAD calls PBGRIB and GRIBEX to read and decode the atmosphere GRIB-formatted forcing data.

11.3.7 Subroutine ATMSTEP

Subroutine ATMSTEP advances the pseudo-atmosphere by one coupled time step. It calculates the heat flux correction based on the mean values of the ocean and atmosphere SST of the previous ’coupled’ time step. It calls ATMREAD to read the forcing fields of the actual ’coupled’ time step. It accumulates the atmospheric output fields and moves the actual mean atmosphere SST to the ’previous’ field. At the end of each month, the mean fields are written onto disk and printed.

11.3.8 Subroutine AVERAGE

Averaging of the accumulated atmospheric forcing fields.

11.3.9 Subroutine COUPINI

Subroutine COUPINI initializes the ocean-atmosphere interface. It sets the default values of parameter and switches and reads the interface NAMELIST KONTCTL. Consistency checks are performed on the date of ocean and atmosphere restart files. Atmosphere and ocean grid specifications are read in. Weight for the coupling are calculated. These are 1 everywhere, if the ocean is globally forced by the atmosphere data. Ocean SST are transferred to the atmosphere. GRIB-formatted atmospheric input fields are opened.

11.3.10 Subroutine INTERPU

Interpolates between ocean and atmosphere grids.

11.3.11 Subroutine OATRANS

Transfer of the ocean SST to the atmosphere by a series of subroutine calls: OGET, INTERPU, APUT.

11.3.12 Subroutine OGET

Transfer of ocean SST from ocean COMMON blocks to the interface.

11.3.13 Subroutine OPUT

Transfer of atmospheric forcing fields to the ocean COMMON block OCEVAL.

11.3.14 Subroutine OPUTRUNO

Transfer of the atmospheric runoff field to the ocean COMMON block OCEVAL. The runoff is distributed equally on the even and odd grids.

11.3.15 Subroutine SETNULL

At the beginning of each ’coupled’ time step, the atmospheric forcing field are initialized in subroutine SETNULL.

12 Appendix B List of ocean model variables

12.1 Parameters

The following parameters have to be specified in PARAM1.h and ZGAUSS.h:

Parameter	Description
IE	number of zonal grid points
JE	number of meridional grid points
KE	number of layers
KBB	bandwidth of the barotropic system matrix
ILL	number of wet oceanic scalar points
NNN720	number of time steps per month
IZGBUFF	reading-buffer size for the barotropic system matrix

12.2 User switches

Switch		Description
IAUFR	1	read restart file
	0	start from rest
IAUFW	1	write a restart file at the end of each month
	0	do not write a restart file
ICESEA	1	sea-ice model active
	0	sea-ice model not active
ICONVA	1	convective adjustment
	0	no convective adjustment
ICYCLI	1	cyclic boundaries
	0	no cyclic boundaries
IDBACK	1	constant vertical background diffusivity
	0	vertically varying background diffusivity (to be specified in subroutine *OCEINI*)
IFWFCORR	1	additional annual mean freshwater flux used

Switch		Description
	0	no additional annual mean freshwater flux used
IEDDY	1	Richardson-number dependent vertical eddy viscosity and diffusivity
	0	constant vertical eddy viscosity and diffusivity
IELIMI	1	direct solution of the barotropic system by Gauss elimination
	0	iterative solution of the barotropic system
IHMIX	1	shear-dependent horizontal diffusion
	0	no shear-dependent horizontal diffusion
IMATW	-1	calculate the matrix, keep it in core memory and run the model
	0	read the matrix from disk at each time step
	1	calculate the matrix, store it on disk and stop
IPME	1	freshwater flux forcing is applied in subroutine OCTHER
	0	no freshwater flux forcing in subroutine OCTHER
IRELSAL	> 0	time scale (days) of salinity relaxation
	0	no salinity relaxation
IRUNO	1	including continental runoff forcing
	0	no continental runoff forcing
ISALRSET	1	reset salt content after sea-level reset
	0	do not reset salt content after sea-level reset
ISLDIFF	1	sea-level diffusion activated
	0	sea-level diffusion not activated
ISLRSET	> 0	reset of mean sea level every *ISLRSET* time steps
	0	no reset of mean sea level
ISOLPR	1	solar penetration active
	0	solar penetration not active
ITSADV	1	thermohaline advection (predictor/corrector) with centered differences
	0	upstream thermohaline advection
JSLRSET	> 0	reset sea level of even/odd grids separately every *JSLRSET* time step
	0	do not reset sea level of even/odd grids separately
NDRUCK		output frequency of print-plots

Switch		Description
NPOLE	1	North Pole included in the model domain
	0	North Pole not included in the model domain
NTOCEAN		no. of ocean steps in one ’coupled’ time step
NTATMOS		no. of atmosphere steps in one ’coupled’ time step
NTAOSTOP		no. of ’coupled’ time steps
NWFREQ		output frequency of diagnostics

12.3 Model constants

Name	Symbol, Unit	Description
ABACK	A_b, [m^2/s]	background vertical viscosity coefficient
ALBW	α_w	surface albedo of water
AUSTAU	A_H, [m^2/s]	horizontal harmonic eddy viscosity coefficient
AV0	A_{V0}, [m^2/s]	coefficient for Richardson-no. dependent vert. eddy viscosity
BHF	$\Delta t B_H/(\Delta x)^4$ []	horizontal eddy viscosity coefficient (biharmonic; subroutine *OCBIHAR*)
BHFR	$\Delta t B_H/(\Delta x)^4$ []	horizontal eddy viscosity coefficient (biharmonic; rotated; subroutine *OCBIHAE*)
BOFRIC	ε, [s^{-1}]	linear bottom friction coefficient
CONN	β	weight of new velocity divergence in continuity equation
CONO	$1-\beta$	weight of old velocity divergence in continuity equation
CRA	C_{RA}	coefficient for Richardson-no. dependent vert. eddy viscosity
CRD	C_{RD}	coefficient for Richardson-no. dependent vert. eddy diffusivity
DBACK	D_b, [m^2/s]	vertical background diffusivity coefficient
DPEN	D, [m]	length scale of solar penetration profile
DH	D_H, [m^2/s]	horizontal harmonic eddy diffusivity coefficient
DSTAB	$-\partial\rho'/\partial z$, [m^{-1}]	stability at which overturning starts (for the definition of $\partial\rho'/\partial z$,see equation (4.1.10))
DT	Δt, [s]	time step

Name	Symbol, Unit	Description
DV0	D_{V0}, [m^2/s]	coefficient for Richardson-no. dependent vertical eddy diffusivity
JGEOO		northernmost row with geostrophic baroclinic mode
JGEOU		southernmost row with geostrophic baroclinic mode
G	g, [m/s^2]	gravitational acceleration of the earth
NSOLPEN		deepest layer with solar penetration
OMEGA	ω, [s^{-1}]	angular velocity of the earth
PI	π	*4.*ATAN(1.)*
RADIUS	*a,* [m]	earth radius
RELAT	λ_{T2}	weight of old horizontal shear-dependent visc./diff. coefficient
RELNE/RELAX	$\lambda_V/(1-\lambda_V)$	weight of new/old Ri.-no.-dependent vertical eddy coefficient
RELTEM	λ_θ, [s^{-1}]	relaxation coefficient for surface temperature
RELSAL	λ_S, [s^{-1}]	relaxation coefficient for surface salinity
RELTI	λ_{T1}	weight of new squared velocity shear in visc./diff. coefficient
ROCON	ρ_{Jacobi}	spectral radius of Jacobi iteration
ROCP	$\rho_o c_p$,[J/m^3K]	volumetric specific heat capacity of sea water
RHOWAT	ρ_o,[kg/m^3]	density of sea water
RPEN	*R*	fraction of solar radiation absorbed in the upper layer
STABEPS	[m^{-1}]	maximum vertical density gradient divided by the mean density with enhanced (m.l.) diffusion/viscosity
STABN	α	weight of new geostrophic components in momentum eq.
STABO	$1-\alpha$	weight of old geostrophic components in momentum eq.
TURDIF	ν_D, [m^2/s]	coefficient for shear-dependent horizontal diffusivity
TURBM		parameter for restriction of shear-dependent diffusion
VISCEQ	ν_A, [m^2/s]	coefficient for shear-dependent horizontal viscosity
VISCM		parameter for restriction of shear-dependent viscosity
WT	W_ρ, [m^2/s]	mixed -layer vertical diffusion/viscosity coefficient
WTDECAY		decay factor for the mixed layer diffusion/viscosity

12.4 Model variables (3-D)

Dimension (*IE,JE,KE+1*)

E/O denotes the EVEN/ODD parity of the north-south index J, i.e. for every array that is indexed by *E/O*, there exist two arrays (e.g. $AV_{E/O}$ stands for arrays *AVE* and *AVO*).

Name	Symbol	Description
$AAV_{E/O}$		work array
$AV_{E/O}$	A_V	vertical Richardson-no. dependent eddy viscosity
$DV_{E/O}$	D_V	vertical Richardson-no. dependent eddy diffusivity
$W_{E/O}$	w	vertical velocity
$WP_{E/O}$	$p'^{n+1,l}$	pressure in baroclinic subsystem iteration

Dimension (*IE,JE,KE)*

Name	Symbol	Description
$AMSU_{E/O}$		'land/sea' mask at vector points (dry=0, wet=1)
$WET_{E/O}$		'land/sea' mask at scalar points (")
$DDU_{E/O}$	d_{uk}	thicknesses of vector grid boxes
$DDP_{E/O}$	d_{wk}	thicknesses of scalar grid boxes
$P_{E/O}$	$p' = p/\rho_o$	relative internal pressure
$S1_{E/O}$		work array
$SA_{E/O}$	S	salinity
$STABI_{E/O}$	$\partial\rho'/\partial z$	negative of vertical density gradient (stability)
$T1_{E/O}$		work array
$TH_{E/O}$	θ	potential temperature
$TURB_{E/O}$	T^2	local squared rate of strain $((\Delta t(v_x + u_y))^2)$
$UK_{E/O}$	u', u	baroclinic zonal velocity / total zonal velocity
$UK1_{E/O}$		work array
$UO_{E/O}$	u	total zonal velocity
$VK_{E/O}$	v', v	baroclinic meridional velocity / total meridional velocity

Name	Symbol	Description
$VK1_{E/O}$		work array
$VO_{E/O}$	v	total meridional velocity

12.5 Model variables (2-D)

Dimension (*IE,JE*)

Name	Symbol	Description
$AREP_{E/O}$		grid cell area at scalar points
$AREU_{E/O}$		grid cell area at vector points
$B1_{E/O}$	Γ_ζ	see equation (6.1.14) on page 35
$DEPT_{E/O}$	H_p, [m]	depth (>0) at scalar points
$DEUT_{E/O}$	H_u, [m]	depth (>0) at vector points
$DEUTI_{E/O}$	$1/H_u$	
$DLX_{E/O}$	$\Delta x_u, \Delta x_\zeta$	zonal distance between 2 vector/scalar points
$DLY_{E/O}$	Δy	meridional distance between 2 vector/scalar points
$DPY_{E/O}$	$\Delta x/\Delta y$	
$DTDXP_{E/O}$	$\Delta t/\Delta x_\zeta$	
$DTDXU_{E/O}$	$\Delta t/\Delta x_u$	
$DTDY_{E/O}$	$\Delta t/\Delta y$	
$DXDY_{E/O}$	$\Delta x_u/\Delta y$	
$DYDX_{E/O}$	$\Delta y/\Delta x_\zeta$	
$FIN_{E/O}$	$1/(1+(\alpha f\Delta t)^2)$	
$FSH_{E/O}$	$(1-\alpha)\, f\Delta t$	
$FTH_{E/O}$	$\alpha f\Delta t$	
$FWFCORR_{E/O}$		annual mean freshwater flux (e.g. correction)
$PX_{E/O}IN$	$\int p'_x dz$	vertically integrated zonal pressure gradient
$PY_{E/O}IN$	$\int p'_y dz$	vertically integrated meridional pressure gradient
$RHELP$		work array (2-D density field)

Name	Symbol	Description
SLOW		work array (2-D salinity field lower layer)
SSUP		work array (2-D salinity field upper layer)
THELP		work array (2-D temperature field)
TLOW		work array (2-D temperature field lower layer)
TSUP		work array (2-D temperature field upper layer)
$U1_{E/O}$	U	vertical integrated zonal velocity (also $(1-\beta)U$)
$US_{E/O}$	$\beta(\Gamma_U + \alpha f \Delta t \Gamma_V)$	see equation (6.1.14)
$UZ_{E/O}$	Γ_U	
$V1_{E/O}$	V	vertical integrated meridional velocity (also $(1-\beta)V$)
$VS_{E/O}$	$\beta(\Gamma_V + \alpha f \Delta t \Gamma_U)$	
$VZ_{E/O}$	Γ_V	
$Z1_{E/O}$	ζ^n	sea level at the old time step
$Z_{E/O}$	ζ^{n+1}	sea level at the new time step

12.6 Model variables (various dimensions)

Name	Dimension	Symbol	Description
AGL	*2 KBB + 1*		part of barotropic system matrix *A*
ABACKV	KE+1	A_b	background vertical viscosity coefficient profile
ALAT	*2 JE*		latitudes
ALONG	*2 IE +6*		longitudes
$AUSTAU_{E/O}$	*JE*	A_H	meridionally varying horizontal viscosity coefficient
B	*ILL + KBB*	b_l^*	right hand side of barotropic system equations after triangularisation
B1	*IE*, 2*JE*	Γ_ζ^n	
$CNUMU_{E/O}$	*KE*		counter of wet vector cells
$CNUMW_{E/O}$	*KE*		counter of wet scalar cells
DBACKV	*KE*+1	D_b	background vertical diffusion coefficient profile
DHI	*JE*	D_H	meridionally varying harmonic diffusion coefficient

Name	Dimension	Symbol	Description
DI	*KE* + 1	$1/\Delta z_u$	
DTVDY	*2 JE*	$\Delta t/\Delta y$	
DWI	*KE*	$1/\Delta z_w$	
DZ	*KE* + 1	Δz_u	vertical distances of horizontal velocity surfaces
DZW	*KE*	Δz_w	vertical distances of vertical velocity surfaces
F	2*JE*	f	Coriolis parameter
H	*IE*, *JE**2	H^*	modified depth (see equation (6.2.2))
HP	*IE*, *JE**2	H_p	total depth at scalar points
HU	*IE*, *JE**2	H_u	total depth at vector points
NUM	*IE*, *JE**2		consecutively numbering of oceanic scalar points
PGL	*2 KBB+1,ILL*	*A*	barotropic system matrix and elimination factors
PREFF	*KE*	p^{ref}	reference pressure
ROREF	*KE*		reference density
SAF	*KE*	[psu]	reference salinity
SKAL	*ILL*	$1/a_{ll}$	scaling factors of the barotropic system matrix (inverse of diagonal elements)
SOLPROF	*KE*		absorption profile for irradiance
TAF	*KE*	[^{o}C]	reference temperature
TIESTU	*KE* + 1	[m]	depth of temperature surfaces
TIESTW	*KE* + 1	[m]	depth of vertical velocity surfaces
TRIDSY	*IE*, *JE*, 3		coefficients of vertical-diffusion/viscosity tridiagonal system
VDY	*2 JE*	Δy	variable meridional distances
VISCEF	*2 JE*	$\nu_A/(\Delta t)$	meridionally varying coefficient for strain-rate dependent viscosity
$WOCSCA_{E/O}$	*JE*	w_r	weights for global/non-global heat flux forcing on scalar points
$WOCVEC_{E/O}$	*JE*		weights for global/non-global wind forcing on vector points
X	*ILL + KBB*	ζ_l^{n+1}	solution vector of barotropic system equations

Name	Dimension	Symbol	Description
ZEITR	*4, NNN720, KE*		time series to be saved on disk

12.7 Arrays used for diagnostics

The arrays marked with a '*' in the following tables are written to the HOPE output file every NWFREQ time step.

Dimension (*IE,JE*)

Name	Unit	Description
$CONVAD_{E/O}$	[]	number of convective adjustment events
$CONVEF_{E/O}$	[10^{-3}W/m^2]	potential energy released by convective adjustment
$CONVOD_{E/O}$	[m]	maximum depth of overturning
$DPM_{E/O}$	[m/s]	total freshwater flux
$PEACC_{E/O}$	[m/s] *	net mean P-E including snow melt/fall
$QDACC_{E/O}$	[W/m^2] *	mean heat flux correction
$QOACC_{E/O}$	[W/m^2] *	mean total heat flux
$ROACC_{E/O}$	[m/s] *	mean run off
$SRACC_{E/O}$	[m/s] *	mean freshtmcdwater flux due to salinity relaxation
$TMOZ_{E/O}$	[m] *	mean sea level
$TXACC_{E/O}$	[Pa] *	mean zonal wind stress (modified by ice/water stress)
$TYACC_{E/O}$	[Pa] *	mean meridional wind stress (")

Dimension (*IE,JE,KE*)

Name	Unit	Description
$TMOT_{E/O}$	[^{o}C] *	mean potential temperature
$TMOS_{E/O}$	[psu] *	mean salinity
$TMOU_{E/O}$	[m/s] *	mean zonal velocity

Name	Unit	Description
$TMOV_{E/O}$	[m/s] *	mean meridional velocity
$TMOW_{E/O}$	[m/s] *	mean vertical velocity
$TMCC_{E/O}$	[ ] *	no. of convective adjustment events
$TMCD_{E/O}$	[m] *	maximum depth of overturning
$TMCE_{E/O}$	$[10^{-3}Wm^{-2}]^*$	mean potential energy released by convective adjustment

12.8 Forcing data and arrays used by the interface/pseudo-atmosphere

Dimension (*IE,JE*)

Name	Symbol, Unit	Description
$AOTX_{E/O}$	τ_a^x, [Pa]	atmospheric zonal wind stress
$AOTY_{E/O}$	τ_a^y, [Pa]	atmospheric meridional wind stress
$AOHFLX_{E/O}$	Q_o, [W/m^2]	atmospheric net heat flux (excluding ice conductive heat flux)
$AOSOL_{E/O}$	Q_s, [W/m^2]	incoming solar radiation
$AODFLX_{E/O}$	Q_Δ, [W/m^2]	atmospheric heat flux correction
$AOTEMP_{E/O}$	[K]	atmospheric 2m-temperature
$AOPME_{E/O}$	*P-E*, [m/s]	atmospheric freshwater flux rate
$AODEWP_{E/O}$	[K]	atmospheric 2m dew-point temperature
$AOSTEM_{E/O}$	[K]	atmospheric SST
$AORUNO_{E/O}$	[m/s]	continental run off
$SAF_{E/O}$	S_{obs}, [psu]	observed upper ocean salinity
$THACC_{E/O}$	[^{o}C]	accumulated ocean SST
$TAF_{E/O}$	θ_{obs}, [^{o}C]	observed mixed layer temperature at actual timestep
$TAF_{E/O}1$	θ_{obs}, [^{o}C]	observed mixed layer temperature at current month
$TAF_{E/O}2$	θ_{obs}, [^{o}C]	observed mixed layer temperature at next month
$TX_{E/O}$	τ_a^x, [Pa]	zonal windstress component at actual time step
$TX_{E/O}1$	τ_a^y, [Pa]	zonal windstress component at current month
$TX_{E/O}2$	τ_a^x, [Pa]	zonal windstress component at next month

Name	Symbol, Unit	Description
$TY_{E/O}$	τ_a^y, [Pa]	meridional windstress component at actual time step
$TY_{E/O}1$	τ_a^y, [Pa]	meridional windstress component at current month
$TY_{E/O}2$	τ_a^y, [Pa]	meridional windstress component at next month

Dimension (*NXATM,NYATM)*

Name	Description
ATACTX	accumulated zonal wind stress on pseudo-atmosphere grid
ATACTY	accumulated meridional wind stress on pseudo-atmosphere grid
ATACPE	accumulated freshwater flux on pseudo-atmosphere grid
ATACSL	accumulated incident solar radiation on pseudo-atmosphere grid
ATACRO	accumulated run off on pseudo-atmosphere grid
ATACQO	accumulated net heat flux on pseudo-atmosphere grid
ATACQD	accumulated heat flux correction on pseudo-atmosphere grid
ATACTO	accumulated surface temperature on pseudo-atmosphere grid
ATACDT	accumulated temperature error on pseudo-atmosphere grid
ATACT2	accumulated 2m temperature on pseudo-atmosphere grid
ATACCL	accumulated cloudiness on pseudo-atmosphere grid
ATACDP	accumulated dew point temperature on pseudo-atmosphere grid
ANEWHEAT	net heat flux of actual atmosphere time step
ANEWSOLAR	incident solar radiation of actual atmosphere time step
ANEWFRESH	freshwater flux of actual atmosphere time step
ANEWTAUX	zonal wind stress of actual atmosphere time step
ANEWTAUY	meridional wind stress of actual atmosphere time step
ANEWSOLUP	upwelling solar radiation of actual atmosphere time step
ANEWTEMP	2m air temperature of actual atmosphere time step
ANEWDEWP	2m dew point temperature of actual atmosphere time step
ANEWCLOU	cloudiness of actual atmosphere time step

Name	Description
ANEWSTEMP	surface temperature of actual atmosphere time step
ANEWRUNO	run-off of actual atmosphere time step
ATEMP	mean 2m air temperature of actual ’coupled’ time step
AHEAT	mean net heat flux of actual ’coupled’ time step
ASOLAR	mean incident solar radiation of actual ’coupled’ time step
AFRESH	mean freshwater flux of actual ’coupled’ time step
ADHEAT	mean relaxation/correction heat flux of actual ’coupled’ time step
ATAUX	mean zonal wind stress of actual ’coupled’ time step
ATAUY	mean meridional wind stress of actual ’coupled’ time step
ACLOU	mean cloudiness of actual ’coupled’ time step
ADEWP	mean 2m dew point temperature of actual ’coupled’ time step
ASTEM	mean surface temperature of actual ’coupled’ time step
ARUNO	mean run-off of actual ’coupled’ time step
WATM	weights for global/non-global coupling; dimension : *NYATM*

NXATM and *NYATM* are the dimensions of the atmospheric model data.

13 Appendix C List of sea-ice model variables

13.1 User switches

Name	Symbol	Description
IOCBRINE	1	brine penetration (subroutine OCBRINE) activated
	0	brine penetration deactivated
ICELEV	> 0	number of thickness categories for thermodynamics
ICESEA	1	sea-ice model activated
	0	sea-ice model not activated
ISNFLG	1	snow layer included
	0	no snow layer included
RELICE	1	salinity relaxation also under the ice-covered part of the grid cell if at all
	0	no salinity relaxation under the ice-covered part of the grid cell

13.2 Model constants

Name	Symbol,Unit	Description
ALBI	α_I	surface albedo of ice
ALBM	α_{IM}	surface albedo of melting ice
ALBSN	α_S	surface albedo of snow
ALBSNM	α_{SM}	surface albedo of melting snow
ARMAX	A_I^{Max}	maximum ice concentration
ARMIN	A_I^{Min}	minimum ice concentration
BRINPROC		rel. part of brine transferred into deeper layers by OCBRINE
CAAPMAX		decay factor for momentum-equation convergence in the Southern Ocean
CAPMAX		decay factor for momentum-equation convergence in the Arctic
CC		specific heat of sea water

Name	Symbol,Unit	Description
CLB	$\rho_I L_f$, [J/m^{3}K]	volumetric heat of fusion of sea ice
CLO		volumetric heat of fusion of snow
CON	κ_I, [W/mK]	conductivity of sea ice
CONSN	κ_S, [W/mK]	conductivity of snow
CW	c_w	ice-water drag coefficient
DHICE	[m^{2}s^{-1}]	coefficient for ice diffusion
D2I	[J/m^{3}K]	coefficient for turbulent heat flux param. over snow or ice
D2W	[J/m^{3}K]	coefficient for turbulent heat flux parameterisation over water
D3	$\varepsilon_{s/w}\sigma$	Stefan Bolzman constant x emissivity of snow/water
HICCE	*e*	principal axes ratio of yield ellipse
HICCP	P^*/ρ_I,[Nm^2/kg]	ice strength coefficient
HSNTOICE	[m]	if the ice thickness exceeds *HSNTOICE*, snow draft is converted into water instead of ice
H0	h_o, [m]	minimum ice thickness of newly created ice
ICEAANB		northern boundary index of the Southern Ocean ice region
ICEAASB		southern boundary index of the Southern Ocean ice region
ICEANB		northern boundary index of Arctic ice region
ICEASB		southern boundary index of Arctic ice region
ITERMAAP		maximum number of iterations for momentum equation in Southern Ocean
ITERMAP		maximum number of iterations for momentum equation in Arctic
OMSOR		underrelaxation coefficient for ice momentum equation solver
PHMAX	[m]	maximum ice thickness in ice strength parameterisation
POMAX		maximum ice concentration in ice strength parameterisation
RHOAIR	ρ_a, [kg/m^3]	density of dry air
RHOICE	ρ_I, [kg/m^3]	density of sea ice
RHOSNO	ρ_S, [kg/m^3]	density of snow
SICE	S_{ice}, [psu]	salinity of sea ice

Name	Symbol,Unit	Description
SINTURN/ COSWAT		sine/cosine of ice-water stress turning angle (not accounted for in the ocean model surface stress!)
SUBL	[J/kg]	latent heat of sublimation
TFREZ	T_{freez}, [^{o}C]	freezing point of sea water
TMELT	T_{melt}, [K]	melting temperature of snow/ice
TREL	[^{o}C]	no salinity relaxation if surface temperature < T_{freez} + *TREL*
VAPL	[J/kg]	latent heat of vaporisation
VRMAXAA		convergence criterion for ice mom. equation in Southern Ocean
VRMAXA		convergence criterion for ice mom. equation in the Arctic

13.3 Model variables (2-D)

Name	Symbol, Unit	Description
$BRINE_{E/O}$	[m]	negative ice thickness change per time step
HDRAFT	h_{draft}, [m]	snow/ice draft
$HIBDEL_{E/O}$	Δ_I, [s^{-1}]	coefficient defining the yield curve of the constitutive law
$HIBET_{E/O}$	η_I, [kg/s]	nonlinear shear viscosity
$HIBZET_{E/O}$	ζ_I, [kg/s]	nonlinear bulk viscosity
$SICOM_{E/O}$	A_I	sea-ice compactness
$SICP_{E/O}$	P_I, [N/m]	sea-ice pressure
$SICSN_{E/O}$	$h_{S,}$ [m]	snow depth
$SICTH_{E/O}$	$h_{I\ ,}$ [m]	sea-ice thickness
$SICU_{E/O}$	$u_{I\ ,}$ [m/s]	zonal sea-ice velocity
$SICV_{E/O}$	$v_{I\ ,}$ [m/s]	meridional sea-ice velocity
$SIOTH_{E/O}$		sea-ice thickness at old time step
$SIOOM_{E/O}$		sea-ice compactness at old time step
$SIOSN_{E/O}$		snow depth at old time step
$TICE_{E/O}$	[^{o}C]	snow/ice skin temperature

13.4 Arrays used for diagnostics

Dimension (*IE,JE*)

Name	Symbol, Unit	Description
$AOACLA_E$	Q_{lat}, [W/m^2]	mean latent heat flux as calculated by the ice model
$AOACLW_E$	Q_l, [W/m^2]	mean longwave heat flux as calculated by the ice model
$AOACSE_E$	Q_{se}, [W/m^2]	mean sensible heat flux as calculated by the ice model
$AOACSW_E$	Q_s, [W/m^2]	mean shortwave heat flux as calculated by the ice model
$FW_{E/O}$	[m/s]	thermodynamic change of ice thickness (m of water)
$PENVAD_{E/O}$		number of brine-penetration events
$PENVOD_{E//O}$	[m]	maximum depth of brine penetration
$PRECN_{E/O}$	[m/s]	net P-E (modified by snow melt/fall)
$QNLA_{E/O}$	[W/m^2]	total latent heat flux as calculated by the ice model
$QNLW_{E/O}$	[W/m^2]	total longwave heat flux as calculated by the ice model
$QNSE_{E/O}$	[W/m^2]	total sensible heat flux as calculated by the ice model
$QNSW_{E/O}$	[W/m^2]	total shortwave heat flux as calculated by the ice model
$RICE_{E/O}$		work array
$TMIZ_{E/O}$	[m] *	mean ice thickness
$TMIU_{E/O}$	[m/s] *	mean zonal ice velocity
$TMIV_{E/O}$	[m/s] *	mean meridional ice velocity
$TMIT_{E/O}$	[^{o}C] *	mean skin temperature
$TMIS_{E/O}$	[m] *	mean snow depth
$TMIC_{E/O}$	[]*	mean ice concentration
$TMGH_{E/O}$	[m/s] *	mean net ice growth

14 Appendix D Code statistics

14.1 Approximate memory requirements

If the barotropic-system matrix is kept on disk, HOPE needs about

- $(44KE + 75)(IE \cdot JE) + 3ILL$ words core memory.

If the barotropic-system matrix is core resident, an additional amount of

- $(2KBB + 1)ILL$ words memory is required.

For the arrays used for diagnostics, an additional amount of

- $(16KE + 24)(IE \cdot JE)$ words core memory is required for the present version.

For the forcing data, the pseudo-atmosphere, and the interface, an additional amount of

- $24(IE \cdot JE) + 35(NXATM \cdot NYATM)$ words is required with daily atmospheric forcing.

For the sea-ice model, the memory requirements are

- $29(IE \cdot JE)$ words of core memory and
- $36(IE \cdot JE)$ words for the diagnostics of the present version.

Additional information on memory requirements for specific models in the pool can be obtained from the example job output that is provided with the models.

14.2 Approximate speed

Information on the CPU time requirements and performance statistics for specific models in the pool can be obtained from the example job output that is provided with the models.

14.3 Emergency hot-line

If you run into problems using HOPE do the following:

1. R.T.F.M. (Read That Fine Manual)
2. check your lines of additional code
3. go to 1.

If you still have problems contact
Stephanie Legutke
legutke@dkrz.de
or
Ernst Maier-Reimer
maier-reimer@dkrz.de.