Formulation and users' guide for Q-GCM. Version 1.2

A. McC. Hogg, J. R. Blundell, W. K. Dewar and P. D. Killworth

May 13, 2004

Contents

1 Introduction

In this document we describe the formulation of an intermediate complexity mid-latitude coupled climate model. The model uses quasi-geostrophic (QG) dynamics in both ocean and atmosphere so that the momentum equations can be solved quickly, while non-linear dynamics and high resolution in the ocean can be retained.

Quasi-geostrophy is known to be a robust approximation to ocean and atmosphere mesoscale dynamics, explicitly involves relative vorticity and eddies, and is much more dynamically transparent than the primitive equations. Further, such a model stands on some twenty odd years of experimentation in ocean-only and atmosphere-only settings and thus represents a logical next step in the continued use of quasi-geostrophy for process modelling. However, QG models are not naturally suited to be run in coupled scenarios because of the poor representation of vertical fluxes of heat, and advection of temperature. An exception is ECBilt (Opsteegh et al., 1998), a 3 layer QG atmosphere model which has been coupled to an ocean model; however the evolution of temperature at the ocean-atmosphere boundary and representation of diabatic processes is oversimplified. Kravtsov and Robertson (2002) describe a coupled QG model which includes an oceanic mixed layer to better represent ocean heat transport. We extend this concept by including mixed layers in both ocean and atmosphere components of a layered QG model. The mixed layers allow the communication of stress and the flux of heat between ocean and atmosphere. The coupling of an atmosphere QG model to an ocean QG model then involves the connection of two symmetric models which use the same equations but different governing parameters. In this way we can model a channel atmosphere coupled to a box ocean, and use this model to simulate a mid-latitude climate system driven by latitudinal variations in incoming radiation. A two layer version of the Quasi-Geostrophic Coupled Model (Q-GCM) is shown schematically in figure 1.

2 Quasi-Geostrophic equations

Begin with the equations for conservation of momentum in a rotating fluid,

$$u_t + uu_x + vu_y + wu_z - fv = -p_x + Ku_{xx} + Ku_{yy} + Ku_{zz}, \tag{2.1a}$$

$$v_t + uv_x + vv_y + wv_z + fu = -p_y + Kv_{xx} + Kv_{yy} + Kv_{zz}, \tag{2.1b}$$

$$w_t + uw_x + vw_y + ww_z = -p_z - g + Kw_{xx} + Kw_{yy} + Kw_{zz}, \tag{2.1c}$$

where we have used $\mathbf{u} = (u, v, w)$ for velocity, and subscripts to represent partial derivatives. The pressure p has been rescaled by mean density ρ and we represent viscosity using a constant diffusion coefficient K. The Coriolis parameter has been represented by f and gravitational acceleration by g. The other equations required here are the continuity equation

$$u_x + v_y + w_z = 0, \tag{2.2}$$

and the mass conservation equation which we write as

$$\rho_t + u\rho_x + v\rho_y + w\rho_z = K_H\rho_{xx} + K_H\rho_{yy} + K_V\rho_{zz}. \tag{2.3}$$

To derive the quasi-geostrophic equations which govern the model dynamics, we simplify (2.1)–(2.3). This could be done rigorously by first non-dimensionalising, and then by scaling away less important terms, but it will suffice for now to do this informally. We begin by assuming, for the purposes of finding the zeroth order equations, that

1. Vertical velocity is small compared with horizontal velocities;

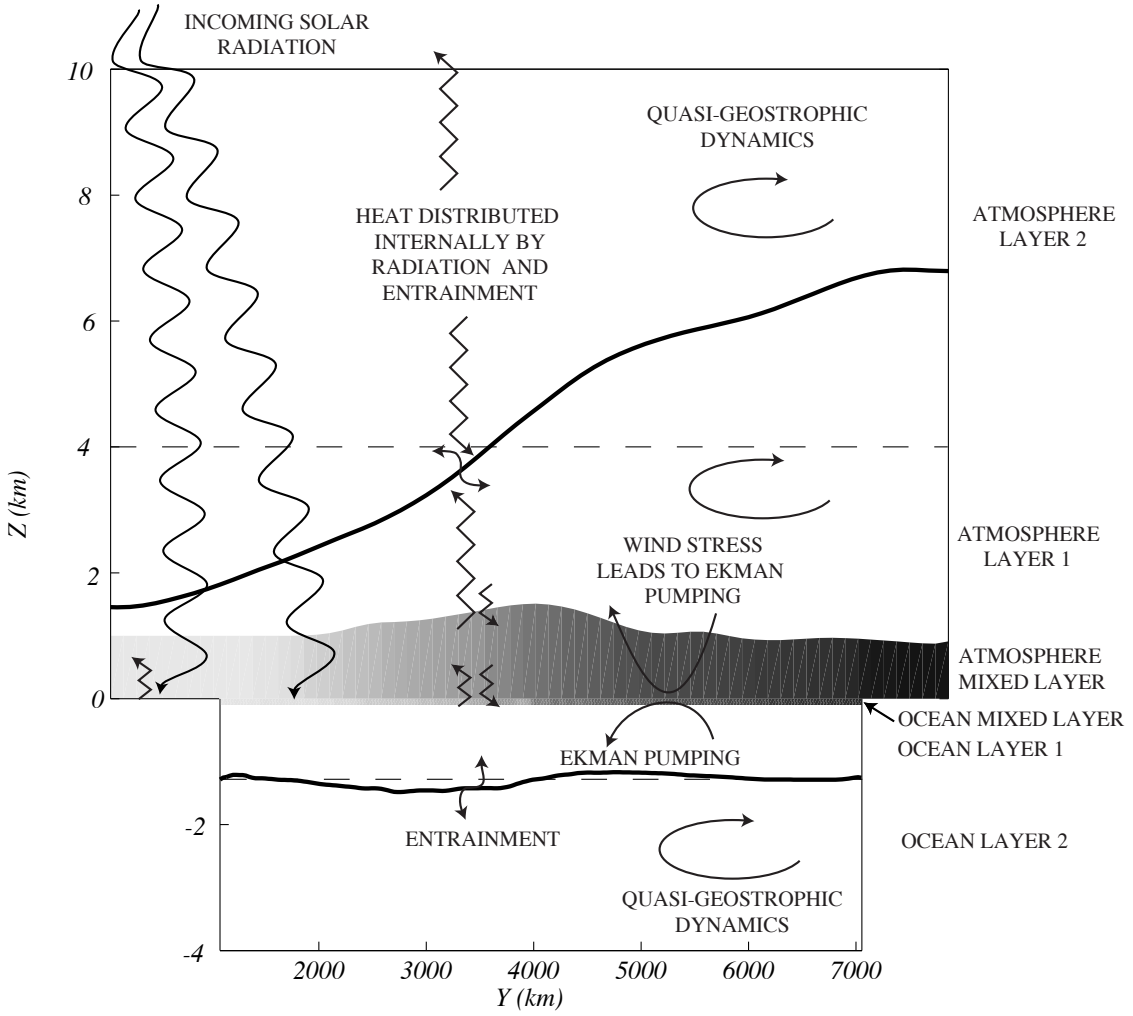

Figure 1: Schematic of (a two-layer version of) the Quasi-Geostrophic Coupled Model. This meridional slice through the model shows the interface dividing the two QG dynamical layers in both the ocean and the atmosphere. The mixed layers, shown by the shading which represents temperature, act to distribute heat and momentum between the two domains. The model is driven by latitudinally varying solar forcing, and by redistribution of heat by longwave radiation in the atmosphere.

2. The Coriolis parameter is represented by the frozen β-plane approximation $f(y) = f_0 + \beta y$ where f_0 represents the zeroth order effect of rotation;

3. Diffusion of momentum is dominated by pressure gradients.

The zeroth order equations in which pressure balances rotation are then

$$f_0 {}^a v_1 = {}^a p_{1x}, \tag{2.4a}$$

$$f_0 {}^a u_1 = -{}^a p_{1y}, \tag{2.4b}$$

$${}^a p_{1z} = -g, \tag{2.4c}$$

and the continuity equation is

$${}^a u_{1x} + {}^a v_{1y} = 0 \tag{2.5}$$

which is automatically satisfied by (2.4a) and (2.4b). Vorticity is given by

$${}^a v_{1x} - {}^a u_{1y} = \frac{\nabla_H^2 {}^a p_1}{f_0}. \tag{2.6}$$

The notation used here indicates that we are referring to layer 1 (subscript 1) in the atmosphere (superscript a). Subscript m is used to represent the mixed layer. The same notation is used for quantities which refer to interface properties, with the ocean-atmosphere boundary defined as interface 0. In the atmosphere (ocean), interface k is defined to be on the upper (lower) boundary of layer k (that is, in both domains we number away from the ocean-atmosphere boundary).

We can now compute the first order quantities (signified by * in the equations below) for the horizontal momentum and continuity equations which include zeroth order horizontal velocity terms and first order pressure and rotation terms,

$$ {}^a u_{1t} + {}^a u_1 {}^a u_{1x} + {}^a v_1 {}^a u_{1y} - f_0 {}^a v_1{}^* - \beta y {}^a v_1 = -{}^a p_{1x}^* - {}^a A_4 \nabla_H^4 {}^a u_1, \tag{2.7a} $$

$$ {}^a v_{1t} + {}^a u_1 {}^a v_{1x} + {}^a v_1 {}^a v_{1y} + f_0 {}^a u_1{}^* + \beta y {}^a u_1 = -{}^a p_{1y}^* - {}^a A_4 \nabla_H^4 {}^a v_1, \tag{2.7b} $$

$$ {}^a u_{1x}^* + {}^a v_{1y}^* + {}^a w_{1z}^* = 0, \tag{2.8} $$

where we have introduced a fourth-order horizontal viscosity scheme (the coefficient ${}^a A_4$ is positive for energy dissipation). First order pressure terms can be eliminated by cross-differentiating (2.7a) and (2.7b), giving

$$ \begin{aligned} {}^a u_{1ty} + ({}^a u_1 {}^a u_{1x})_y + ({}^a v_1 {}^a u_{1y})_y - f_0 {}^a v_{1y}^* - (\beta y {}^a v_1)_y + {}^a A_4 \nabla_H^4 {}^a u_{1y} \\ = {}^a v_{1tx} + ({}^a u_1 {}^a v_{1x})_x + ({}^a v_1 {}^a v_{1y})_x + f_0 {}^a u_{1x}^* + (\beta y {}^a u_1)_x + {}^a A_4 \nabla_H^4 {}^a v_{1x}, \end{aligned} \tag{2.9} $$

and rearranging:

$$ \begin{aligned} \left(\frac{\nabla_H^2 {}^a p_1}{f_0} + \beta y \right)_t + \left({}^a u_1 \left[\frac{\nabla_H^2 {}^a p_1}{f_0} + \beta y \right] \right)_x + \left({}^a v_1 \left[\frac{\nabla_H^2 {}^a p_1}{f_0} + \beta y \right] \right)_y + \\ ({}^a u_{1x} + {}^a v_{1y}) \left(\frac{\nabla_H^2 {}^a p_1}{f_0} + \beta y \right) + f_0 ({}^a u_{1x}^* + {}^a v_{1y}^*) = -{}^a A_4 \nabla_H^4 \left(\frac{\nabla_H^2 {}^a p_1}{f_0} \right) \end{aligned} \tag{2.10} $$

Insert (2.5) and (2.8) to eliminate zeroth order divergence terms and first order velocities, leaving an equation which depends only upon zeroth order terms

$$ \begin{aligned} \left(\frac{\nabla_H^2 {}^a p_1}{f_0} + \beta y \right)_t + \left({}^a u_1 \left[\frac{\nabla_H^2 {}^a p_1}{f_0} + \beta y \right] \right)_x + \left({}^a v_1 \left[\frac{\nabla_H^2 {}^a p_1}{f_0} + \beta y \right] \right)_y = \\ f_0 {}^a w_{1z}^* - \frac{{}^a A_4}{f_0} \nabla_H^6 {}^a p_1. \end{aligned} \tag{2.11} $$

We now integrate (2.11) over the layer depth. Note that it is assumed that ${}^a p_1$, ${}^a u_1$ and ${}^a v_1$ are effectively layer-averaged quantities, so that integration between zero and ${}^a H_1$ simply gives

$$ \begin{aligned} \left(\frac{\nabla_H^2 {}^a p_1}{f_0} \right)_t + \left({}^a u_1 \frac{\nabla_H^2 {}^a p_1}{f_0} \right)_x + \left({}^a v_1 \left(\frac{\nabla_H^2 {}^a p_1}{f_0} + \beta y \right) \right)_y = \\ \frac{f_0 ({}^a w_1{}^* - {}^a w_m{}^*)}{{}^a H_1} - \frac{{}^a A_4}{f_0} \nabla_H^6 {}^a p_1, \end{aligned} \tag{2.12} $$

which is an equation for the evolution of potential vorticity in the layer. Evaluation of vertical velocity at the layer boundaries (${}^a w_1{}^*$ and ${}^a w_m{}^*$) requires knowledge of the flux between layers (which we call e for net entrainment velocity through the boundary of the layer). Vertical velocity is a first order quantity and thus incorporates variable layer thickness ${}^a h_1$, giving

$$ {}^a w_1{}^* = {}^a h_{1t} + ({}^a u_1 {}^a h_1)_x + ({}^a v_1 {}^a h_1)_y + {}^a e_1. \tag{2.13} $$

The lower boundary of the atmosphere is solid; however vertical velocity due to Ekman pumping (derived below) makes a contribution; ${}^a w_m{}^* = {}^a w_{ek}$. We can now write an equation for the lowest atmospheric layer dynamics,

$$ {}^a q_{1t} + ({}^a u_1 {}^a q_1)_x + ({}^a v_1 {}^a q_1)_y = \frac{f_0({}^a e_1 - {}^a w_{ek})}{{}^a H_1} - \frac{{}^a A_4}{f_0} \nabla_H^6 {}^a p_1, \tag{2.14} $$

where layer potential vorticity is defined as

$$ {}^a q_1 \equiv \frac{\nabla_H^2 {}^a p_1}{f_0} + \beta y - \frac{f_0 {}^a \eta_1}{{}^a H_1} + \frac{f_0 {}^a D}{{}^a H_1}, \tag{2.15} $$

and the perturbation interface height, ${}^a \eta_1 \equiv {}^a h_1 - {}^a H_1 = \frac{{}^a p_1 - {}^a p_2}{{}^a g_1'}$. Here we have introduced the bottom topography ${}^a D(x,y)$ which is assumed to be small compared to layer thickness ${}^a H_1$, for consistency with the quasigeostrophic scaling. Equivalent equations can be derived for each layer in the model. For other layers in the N-layer atmosphere these are:

$$ {}^a q_{kt} + ({}^a u_k {}^a q_k)_x + ({}^a v_k {}^a q_k)_y = \frac{f_0}{{}^a H_k}({}^a e_k - {}^a e_{k-1}) - \frac{{}^a A_4}{f_0} \nabla_H^6 {}^a p_k, \tag{2.16} $$

where ${}^a e_N \equiv 0$, and

$$ {}^a q_k \equiv \frac{\nabla_H^2 {}^a p_k}{f_0} + \beta y - \frac{f_0}{{}^a H_k}({}^a \eta_k - {}^a \eta_{k-1}), \qquad \text{for } k = 2, N, \tag{2.17} $$

$$ {}^a \eta_k = \frac{{}^a p_k - {}^a p_{k+1}}{{}^a g_k'}, \qquad \text{for } k = 1, N-1, \tag{2.18} $$

$$ {}^a \eta_N \equiv 0. \tag{2.19} $$

For an N-layer ocean, the equations are

$$ {}^o q_{kt} + ({}^o u_k {}^o q_k)_x + ({}^o v_k {}^o q_k)_y = \frac{f_0}{{}^o H_k}({}^o e_{k-1} - {}^o e_k) + \frac{{}^o A_2}{f_0} \nabla_H^4 {}^o p_k - \frac{{}^o A_4}{f_0} \nabla_H^6 {}^o p_k, \qquad \text{for } k = 1, N \tag{2.20} $$

where a Laplacian diffusion term is included, the vertical velocities at the top and bottom of the ocean are given by ${}^o e_0 \equiv {}^o w_{ek}$ and ${}^o e_N \equiv \frac{\delta_{ek}}{2 f_0} \nabla_H^2 {}^o p_N$ as will be discussed in §3.1, and

$$ {}^o q_k \equiv \frac{\nabla_H^2 {}^o p_k}{f_0} + \beta y + \frac{f_0}{{}^o H_k}({}^o \eta_k - {}^o \eta_{k-1}), \qquad \text{for } k = 1, N-1, \tag{2.21} $$

$$ {}^o q_N \equiv \frac{\nabla_H^2 {}^o p_N}{f_0} + \beta y - \frac{f_0}{{}^o H_N} {}^o \eta_{N-1} + \frac{f_0 {}^o D}{{}^o H_N}, \tag{2.22} $$

$$ {}^o \eta_k = -\frac{{}^o p_k - {}^o p_{k+1}}{{}^o g_k'}, \qquad \text{for } k = 1, N-1, \tag{2.23} $$

$$ {}^o \eta_0 \equiv 0. $$

All of the model equations can be written more generally in matrix form

$$ \mathbf{q}_t + (\mathbf{uq})_x + (\mathbf{vq})_y = \mathsf{Be} + \frac{A_2}{f_0} \nabla_H^4 \mathbf{p} - \frac{A_4}{f_0} \nabla_H^6 \mathbf{p}, \tag{2.24} $$

$$ f_0 \mathbf{q} - f_0 \beta y = \nabla_H^2 \mathbf{p} - f_0^2 \mathsf{A} \mathbf{p} + f_0 \tilde{\mathbf{D}}, \tag{2.25} $$

where $\tilde{\mathbf{D}}$ is the dynamic topography which is rescaled by f_0 and the inverse of layer thickness (see 5.8, 5.9), ${}^a A_2 \equiv 0$, A is an $N \times N$ matrix containg the coefficients of the η contribution to

vorticity, which has the same structure for the ocean and atmosphere, while B is an $N \times N+1$ matrix containg the coefficients of the forcing terms, which differs only by a sign between the two cases (see §5.1 for details of the matrices and vectors). Note that $\mathbf{e}$ is a vector of length $N+1$ which is yet to be determined.

The boundary conditions in the model are mixed conditions following Haidvogel et al. (1992). The conditions are applied to the pressure field, and are written for the example of the north and south boundaries of the atmosphere to be

$$ {}^{a}p_k = {}^{a}f_k(t), \tag{2.26} $$

$$ {}^{a}p_{knn} = -\frac{{}^{a}\alpha_{bc}}{\Delta}{}^{a}p_{kn}, \tag{2.27} $$

$$ {}^{a}p_{k4n} = -\frac{{}^{a}\alpha_{bc}}{\Delta}{}^{a}p_{k3n}, \tag{2.28} $$

where the dimensionless coefficient ${}^{a}\alpha_{bc}$ is zero for free slip and large for no-slip boundary conditions, and subscript n denotes the outward normal derivative. The function ${}^{a}f_k(t)$ is determined using constraints on mass and momentum, after McWilliams (1977), and will be different on the northern and southern boundaries in the atmosphere depending upon the net flow in the layer. These conditions are derived in appendix B.2. The east and west boundaries are periodic. The ocean boundaries are treated using the same mixed boundary conditions described by (2.26)–(2.28), with the x-derivative of pressure used on the east and west boundaries. The value of ${}^{o}f_k(t)$ is constrained by mass conservation and is the same for all boundaries.

3 Mixed layers

Quasi-geostrophy describes, to first order, the dynamics of both ocean and atmosphere, but cannot handle the fluxes of heat and momentum between these two domains which are required for a coupled model. Q-GCM avoids this difficulty by embedding mixed layers within layer 1 of both ocean and atmosphere. In its current form the mixed layers do not include parameterisation of humidity in the atmosphere or salinity in the ocean, and therefore exchanges of these quantities are not modelled. Nonetheless, the most fundamental processes – transfer of momentum from atmosphere to ocean, and flux of heat in both directions – are included.

3.1 Mixed layer stress

We first derive the stress at the ocean-atmosphere interface by finding the atmospheric mixed layer velocities. The distinction between mixed layer and layer 1 velocities is that the mixed layer is thin enough that the vertical flux of horizontal momentum is significant. The zeroth order momentum equations can be found by including the vertical flux of momentum by a dynamic stress (rescaled by mean density ${}^{a}\rho$)

$$ -f_0{}^{a}v_m = -{}^{a}p_{1_x} + {}^{a}\tau_z^x, \tag{3.1a} $$

$$ f_0{}^{a}u_m = -{}^{a}p_{1_y} + {}^{a}\tau_z^y. \tag{3.1b} $$

Assuming a fixed atmospheric mixed layer thickness and zero stress at the upper boundary of the mixed layer, we substitute (2.4a,b) and integrate over mixed layer depth ${}^{a}H_m$ to give

$$ {}^{a}v_m = {}^{a}v_1 + \frac{{}^{a}\tau^x}{{}^{a}H_m f_0}, \tag{3.2a} $$

$$ {}^{a}u_m = {}^{a}u_1 - \frac{{}^{a}\tau^y}{{}^{a}H_m f_0}, \tag{3.2b} $$

where it is implied that stress is a layer-averaged quantity. We assume stress is a quadratic function of the ageostrophic mixed layer velocity (Pedlosky, 1987),

$$({}^a\tau^x, {}^a\tau^y) = C_D|\mathbf{^a u_m}|({}^a u_m, {}^a v_m), \tag{3.3}$$

C_D being a dimensionless drag coefficient. Substituting (3.3) into (3.2a,b) gives

$${}^a v_m = {}^a v_1 + \frac{C_D|\mathbf{^a u_m}|{}^a u_m}{{}^a H_m f_0}, \tag{3.4a}$$

$${}^a u_m = {}^a u_1 - \frac{C_D|\mathbf{^a u_m}|{}^a v_m}{{}^a H_m f_0}. \tag{3.4b}$$

which allows us to write, with some algebraic manipulation,

$${}^a v_m = \left[{}^a v_1 + \frac{C_D|\mathbf{^a u_m}|{}^a u_1}{{}^a H_m f_0}\right]\left[1 + \left(\frac{C_D|\mathbf{^a u_m}|}{{}^a H_m f_0}\right)^2\right]^{-1}, \tag{3.5a}$$

$${}^a u_m = \left[{}^a u_1 - \frac{C_D|\mathbf{^a u_m}|{}^a v_1}{{}^a H_m f_0}\right]\left[1 + \left(\frac{C_D|\mathbf{^a u_m}|}{{}^a H_m f_0}\right)^2\right]^{-1}, \tag{3.5b}$$

allowing direct calculation of stress provided one knows

$$|\mathbf{^a u_m}| \equiv \sqrt{{}^a u_m{}^2 + {}^a v_m{}^2} = \frac{{}^a H_m f_0}{\sqrt{2}\, C_D}\sqrt{-1 + \sqrt{1 + 4\left(\frac{C_D|\mathbf{^a u_1}|}{{}^a H_m f_0}\right)^2}}. \tag{3.6}$$

A second distinction between mixed layer and layer 1 velocities exists because the ageostrophic mixed layer velocities have a divergent component which produces vertical Ekman velocities. Ekman pumping is calculated using the continuity equation (2.2) integrated from 0 to ${}^a H_m$,

$${}^a w_{ek} = -{}^a H_m{}^a u_{mx} - {}^a H_m{}^a v_{my}, \tag{3.7}$$

where we use ${}^a w_m(0) = 0$ and ${}^a w_m({}^a H_m) = {}^a w_{ek}$. Substitute velocity from (3.2a,b), giving

$${}^a w_{ek} = \frac{{}^a\tau_x^y - {}^a\tau_y^x}{f_0}, \tag{3.8}$$

the familiar expression for Ekman velocity as a function of stress.

The ocean mixed layer is essentially an inversion of the atmospheric mixed layer. The difference here is that stress in the oceanic mixed layer is passive, depending entirely on wind stress and the atmosphere to ocean density ratio,

$${}^o\tau = \xi\frac{{}^a\rho}{{}^o\rho}{}^a\tau, \tag{3.9}$$

where ξ is an order 1 stress multiplier and is designed to take into account the poor vertical resolution of the atmosphere in the boundary layer. Ekman pumping in the ocean surface layer can be calculated from stress in the same way as (3.8) to give

$${}^o w_{ek} = \frac{{}^o\tau_x^y - {}^o\tau_y^x}{f_0}. \tag{3.10}$$

It is relevant at this stage to note that momentum must also be lost from the ocean, and that this occurs via drag due to a no-slip condition at the ocean floor. We calculate this by assuming an Ekman layer of prescribed thickness $\delta_{ek} = \sqrt{K/f_0}$ and proceed as above. In this case there

is no need to calculate stress explicitly, and so we determine the Ekman pumping directly from pressure. Velocity in the Ekman layer will be given by

$$ {}^{o}v_N = \frac{{}^{o}p_{Nx}}{f_0} + \frac{\tau^x}{\delta_{ek} f_0}, \tag{3.11a} $$

$$ {}^{o}u_N = -\frac{{}^{o}p_{Ny}}{f_0} - \frac{\tau^y}{\delta_{ek} f_0}. \tag{3.11b} $$

We use linear stress,

$$ (\tau^x, \tau^y) = \frac{K}{\delta_{ek}} ({}^{o}u_N, {}^{o}v_N), \tag{3.12} $$

so that

$$ {}^{o}v_N = \frac{{}^{o}p_{Nx}}{f_0} + {}^{o}u_N, \tag{3.13a} $$

$$ {}^{o}u_N = -\frac{{}^{o}p_{Ny}}{f_0} - {}^{o}v_N. \tag{3.13b} $$

Combining these two we can write separate equations for each component of velocity,

$$ {}^{o}v_N = \frac{{}^{o}p_{Nx}}{2f_0} - \frac{{}^{o}p_{Ny}}{2f_0}, \tag{3.14a} $$

$$ {}^{o}u_N = -\frac{{}^{o}p_{Ny}}{2f_0} - \frac{{}^{o}p_{Nx}}{2f_0}. \tag{3.14b} $$

These can be substituted into (2.2), which is then integrated across the Ekman layer:

$$ {}^{o}w_N = \frac{\delta_{ek}}{2f_0} \nabla_H^2 {}^{o}p_N. \tag{3.15} $$

This vertical velocity on the lower boundary of the ocean alters the vertical velocity in ocean layer N but has no impact on the heat flux (as there is no temperature gradient).

3.2 Mixed layer evolution

The mixed layers also act to transfer heat between atmosphere and ocean. To allow this, we need to explicitly calculate the evolution of temperature in the mixed layers. As before we integrate over the mixed layer thickness, but for the purposes of temperature evolution we allow perturbations ${}^{a}\eta_m = {}^{a}h_m - {}^{a}H_m$ to mixed layer height in the atmosphere only. The variable mixed layer height is needed in the atmosphere because of an observed instability which occurs due to large vertical heat transports in regions where there is a coherent pattern of upwelling and cold mixed layer temperatures.

To find the evolution of mixed layer height, we again integrate the continuity equation, this time from 0 to ${}^{a}h_m$,

$$ {}^{a}w_m({}^{a}h_m) = -{}^{a}h_m {}^{a}u_{mx} - {}^{a}h_m {}^{a}v_{my}. \tag{3.16} $$

Now we write an equation for the mixed layer height, analogous to (2.13),

$$ {}^{a}w_m({}^{a}h_m) = {}^{a}h_{mt} + {}^{a}u_m {}^{a}h_{mx} + {}^{a}v_m {}^{a}h_{my} + {}^{a}e_m. \tag{3.17} $$

Combining these gives the evolution equation for mixed layer height, which implicitly includes the Ekman divergence terms,

$$ {}^{a}h_{mt} + ({}^{a}u_m {}^{a}h_m)_x + ({}^{a}v_m {}^{a}h_m)_y = -{}^{a}e_m. \tag{3.18} $$

This equation is solved with ${}^{a}h_m = {}^{a}H_m$ on the north and south boundaries.

The evolution of mixed layer temperature comes from a heat equation,

$$ {}^aT_{mt} + {}^au_m\,{}^aT_{mx} + {}^av_m\,{}^aT_{my} + {}^aw_m\,{}^aT_{mz} = {}^aK_2\nabla_H^2\,{}^aT_m - {}^aK_4\nabla_H^4\,{}^aT_m - \frac{{}^aF_z}{{}^a\rho\,{}^aC_p}, \tag{3.19} $$

where ${}^aC_p$ is the specific heat capacity and ${}^aK_2$ a diffusion coefficient. This equation includes a fourth-order diffusion term with positive coefficient ${}^aK_4$ for numerical stability. We have chosen to retain the horizontal diffusion terms, while writing the vertical diffusion as the vertical gradient of a flux ${}^aF$ of heat energy. Note that flux is defined positive upwards so that a positive flux gradient will act to cool the layer in a stably stratified fluid. Integrate (3.19) between 0 and ${}^ah_m$,

$$ {}^aT_{mt} + ({}^au_m\,{}^aT_m)_x + ({}^av_m\,{}^aT_m)_y + \frac{{}^aw_{ek}\,{}^aT_m}{{}^aH_m} = {}^aK_2\nabla_H^2\,{}^aT_m - {}^aK_4\nabla_H^4\,{}^aT_m + \frac{-{}^aF_m + {}^aF_0}{{}^a\rho\,{}^aC_p\,{}^ah_m}. \tag{3.20} $$

The fluxes at the top of the mixed layer (${}^aF_m$) and at the surface (${}^aF_0$) are derived in §4 below. There is no heat flux on the north and south boundaries.

In the ocean, we use a constant thickness mixed layer, and the equivalent equation to (3.20) is

$$ {}^oT_{mt} + ({}^ou_m\,{}^oT_m)_x + ({}^ov_m\,{}^oT_m)_y - \frac{{}^ow_{ek}\,{}^oT_m}{{}^oH_m} = {}^oK_2\nabla_H^2\,{}^oT_m - {}^oK_4\nabla_H^4\,{}^oT_m + \frac{-{}^oF_0 + {}^oF_m}{{}^o\rho\,{}^oC_p\,{}^oH_m}. \tag{3.21} $$

This is solved with zero heat flux on east, west and north boundaries. The southern boundary condition uses ${}^oT_m = {}^oT_S$, a constant determined from radiative balance. This condition is necessary because advection and diffusion of warm tropical water from the south are not explicitly included in the model. A similar boundary condition is not required in the north because convection is active there.

4 Flux between layers

Q-GCM includes flux of heat between each layer due to a number of different processes. We derive each case below, and write heat flux as the superposition of a mean and a first order perturbation quantity, so that fluxes are linearised about a mean state in which the model is in radiative balance.

4.1 Incoming solar flux

The incoming solar radiation is written as a heat flux $F_S = \overline{F_s} + F_s'(y)$. The atmosphere is transparent to incoming shortwave radiation which is deposited directly in the oceanic mixed layer. Over land, incoming shortwave is absorbed by the land and immediately re-emitted as longwave radiation into the atmospheric mixed layer. Heat flux is defined so that positive fluxes denote an upwards movement of heat energy, so that $\overline{F_s}$ is defined negative. We choose the perturbation to radiation so that it integrates to zero over the domain,

$$ F_s'(y) = -\left|F_s'\right| \cos\left(\frac{\pi y}{{}^aY}\right), \tag{4.1} $$

where ${}^aY$ is the latitudinal atmospheric domain size.

4.2 Radiative fluxes

Each atmospheric layer, and the oceanic mixed layer, radiates and absorbs heat depending upon its temperature and optical depth. The emitted longwave radiation is either absorbed, transmitted or partially transmitted by neighbouring layers. The equations describing radiative flux from each layer are given below, in the form of a mean part plus linear perturbations. The constant coefficients governing the linear response of the model are derived in appendix A.

Oceanic Mixed Layer absorbs all downgoing longwave radiation (as well as incoming shortwave radiation) which passes through the atmospheric mixed layer. Emits radiation according to

$$F_0^{\uparrow} \approx \overline{F_0^{\uparrow}} + D_0^{\uparrow o}T_m{}'. \tag{4.2}$$

Atmospheric Mixed Layer absorbs all downgoing and upgoing longwave radiation. Linearised radiation is written

$$F_m^{\uparrow} \approx \overline{F_m^{\uparrow}} + B_m^{\uparrow\,a}\eta_m + C_m^{\uparrow\,a}D + D_m^{\uparrow\,a}T_m{}', \tag{4.3}$$

$$F_m^{\downarrow} \approx \overline{F_m^{\downarrow}} + D_m^{\downarrow\,a}T_m{}'. \tag{4.4}$$

Atmospheric QG layers have a more complicated radiation balance, as partial absorption of incoming radiation and partial emission of the incoming radiation occur:

$$F_k^{\uparrow} \approx \overline{F_k^{\uparrow}} + \sum_{i=1}^{\min(k,N-1)} A_{k,i}^{\uparrow\,a}\eta_i + B_k^{\uparrow a}\eta_m + C_k^{\uparrow a}D + D_k^{\uparrow a}T_m{}' \qquad \text{for } k = 1, N, \tag{4.5}$$

$$F_k^{\downarrow} \approx \overline{F_k^{\downarrow}} + \sum_{i=\max(k-1,1)}^{N-1} A_{k,i}^{\downarrow\,a}\eta_i + B_k^{\downarrow a}\eta_m + C_k^{\downarrow a}D \qquad \text{for } k = 1, N. \tag{4.6}$$

4.3 Sensible and latent heat flux

Sensible and latent heat flux from the oceanic to the atmospheric mixed layer is represented by

$$F_\lambda = \lambda({}^oT_m - {}^aT_m), \tag{4.7}$$

or in perturbation form

$$F_\lambda = \overline{F_\lambda} + \lambda({}^oT_m{}' - {}^aT_m{}'), \tag{4.8}$$

where

$$\overline{F_\lambda} = \lambda(\overline{{}^oT_m} - \overline{{}^aT_m}). \tag{4.9}$$

4.4 Entrainment heat flux

The QG layers have constant (but different) potential temperatures. Therefore, the combined heat content of any two layers can only change when the interface dividing the layers changes its height. The entrainment ${}^ae_i$ therefore implies a vertical heat flux into or out of layers i and $i+1$. This heat flux is written as a double-sided flux (see McDougall and Dewar, 1998), so that in the atmosphere

$${}^aF_i^{e+} - {}^aF_i^{e-} = -{}^a\rho^a C_p\,{}^a e_i \Delta_i^a T, \tag{4.10}$$

where $\Delta_i^a T$ is the change in temperature across the interface and ${}^aF_i^{e+}$ (${}^aF_i^{e-}$) is the heat flux on the upper (lower) side of interface i. This allows a divergence of heat flux at the interface, but not within the layers.

In the atmosphere we assume no entrainment at the top of the domain:

$${}^aF_N^{e-} = 0. \tag{4.11}$$

The heat conservation equation in any layer is

$${}^aF_i^{e-} + F_i^{\uparrow} + F_{i+1}^{\downarrow} = {}^aF_{i-1}^{e+} + F_{i-1}^{\uparrow} + F_i^{\downarrow}, \tag{4.12}$$

allowing calculation of entrainment at the interface i by combining the equations for layers i and $i+1$;

$$ {}^a e_i = \frac{{}^a F_{i-1}^{e+} + F_{i-1}^{\uparrow} + F_i^{\downarrow} - {}^a F_{i+1}^{e-} - F_{i+1}^{\uparrow} - F_{i+2}^{\downarrow}}{{}^a\rho^a C_p \Delta_i^a T}. \tag{4.13} $$

In the 2 layer case this equation is conclusive, since ${}^a F_m^{e+} = 0$ (see below) and ${}^a F_N^{e-} = 0$ (because it is at the top of the atmosphere), so that we can write the equation for ${}^a e_1$ explicitly. For more layers we have insufficient constraints to determine entrainment everywhere. Therefore, on the basis that most convection and vertical heat transport occurs in the lower atmosphere, we assume that all entrainment occurs at the lowest interface, or in other words

$$ {}^a e_1 = \frac{F_m^{\uparrow} + F_1^{\downarrow} - F_N^{\uparrow}}{{}^a\rho^a C_p \Delta_1^a T}. \tag{4.14} $$

$$ {}^a e_i = 0, \qquad i > 1. \tag{4.15} $$

All radiation in the atmosphere is therefore encapsulated in (4.14), which we now rewrite explicitly in terms of the radiation parameters used in (4.3), (4.5) and (4.6):

$$ {}^a e_1 = \frac{1}{{}^a\rho^a C_p \Delta_1^a T}\Bigg[\sum_{i=1}^{N-1}\left(A_{1,i}^{\downarrow} - A_{N,i}^{\uparrow}\right){}^a\eta_1 + \left(B_m^{\uparrow} + B_1^{\downarrow} - B_N^{\uparrow}\right){}^a\eta_m + \left(C_m^{\uparrow} + C_1^{\downarrow} - C_N^{\uparrow}\right){}^a D + \left(D_m^{\uparrow} - D_N^{\uparrow}\right){}^a T_m{}'\Bigg]. \tag{4.16} $$

At the mixed layer interface we assume that entrainment is maintained by turbulence on either side of the interface. In other words, assume

$$ {}^a F_m^{e+} = \overline{F_x} - \frac{{}^a\rho^a C_p{}^a\mu_1}{{}^a H_1 - {}^a h_m}, \tag{4.17} $$

$$ {}^a F_m^{e-} = \overline{F_x} - \frac{{}^a\rho^a C_p{}^a\mu_m}{{}^a h_m}, \tag{4.18} $$

where $\overline{F_x}$ is an export heat flux (a constant added heat flux), and

$$ \frac{{}^a\mu_1}{{}^a H_1 - {}^a H_m} = \frac{{}^a\mu_m}{{}^a H_m}. \tag{4.19} $$

We linearise the above equations to give

$$ {}^a F_m^{e+} = \overline{F_x} - \frac{{}^a\rho^a C_p{}^a\mu_1}{{}^a H_1 - {}^a H_m}, \tag{4.20} $$

$$ {}^a F_m^{e-} = \overline{{}^a F_m^{e-}} - \frac{{}^a\rho^a C_p{}^a\mu_m}{{}^a H_m}\left(1 - \frac{{}^a\eta_m}{{}^a H_m}\right). \tag{4.21} $$

Then

$$ {}^a e_m \approx \frac{{}^a\phi_m}{{}^a\rho^a C_p \Delta_m^a T}{}^a\eta_m, \tag{4.22} $$

where

$$ {}^a\phi_m \equiv \frac{{}^a\rho^a C_p{}^a\mu_m}{{}^a H_m{}^2}. \tag{4.23} $$

In the model we then use the perturbation forcing

$$ {}^a F_m^{e-} = \overline{{}^a F_m^{e-}} + {}^a\phi_m{}^a\eta_m \tag{4.24} $$

$$ {}^{a}F_m^{e+} = \overline{{}^{a}F_m^{e-}}. \tag{4.25}$$

In the ocean we assume that the deep layers do not exchange heat so that

$$ {}^{o}F_1^{e-} = 0, \tag{4.26}$$

and heat flux between other layers is zero,

$$ {}^{o}e_k = 0, \qquad k > 1 \tag{4.27}$$

From (4.10) we write

$$ {}^{o}F_1^{e+} = -{}^{o}\rho {}^{o}C_p {}^{o}e_1 \Delta_1^o T. \tag{4.28}$$

In the oceanic mixed layer, where there is no variation in mixed layer depth, entrainment is solely driven by Ekman pumping,

$$ {}^{o}e_m = {}^{o}w_{ek}. \tag{4.29}$$

Using a centred difference model for entrainment there, we find

$$ {}^{o}F_m^{e+} = -0.5{}^{o}\rho {}^{o}C_p \Delta_m^o T {}^{o}w_{ek}, \tag{4.30}$$

$$ {}^{o}F_m^{e-} = 0.5{}^{o}\rho {}^{o}C_p \Delta_m^o T {}^{o}w_{ek}. \tag{4.31}$$

Therefore we can write

$$ {}^{o}e_1 = -\frac{\Delta_m^o T}{2\Delta_1^o T} {}^{o}w_{ek} \tag{4.32}$$

5 Layer equations

We are now in a position to put together equations for each layer. A schematic of the model which includes radiative fluxes, entrainment velocities and momentum coupling is shown in figure 1.

5.1 QG layers

The QG equations are written from (2.24) and (2.25) as follows,

$$ \mathbf{q}_t + (\mathbf{uq})_x + (\mathbf{vq})_y = \mathsf{Be} + \frac{A_2}{f_0}\nabla_H^4 \mathbf{p} - \frac{A_4}{f_0}\nabla_H^6 \mathbf{p}, \tag{5.1}$$

which can also be written as

$$ \mathbf{q}_t = \frac{1}{f_0} J(\mathbf{q}, \mathbf{p}) + \mathsf{Be} + \frac{A_2}{f_0}\nabla_H^4 \mathbf{p} - \frac{A_4}{f_0}\nabla_H^6 \mathbf{p}, \tag{5.2}$$

where J is the Jacobian,

$$ J(\mathbf{q}, \mathbf{p}) = \mathbf{q}_x \mathbf{p}_y - \mathbf{q}_y \mathbf{p}_x, $$

and

$$ f_0 \mathbf{q} - f_0 \beta y = \nabla_H^2 \mathbf{p} - f_0^2 \mathsf{A}\mathbf{p} + f_0 \tilde{\mathbf{D}}. \tag{5.3}$$

For both ocean and atmosphere we can write

$$ \mathsf{A} = \begin{bmatrix} \frac{1}{H_1 g_1'} & \frac{-1}{H_1 g_1'} & & & \\ \frac{-1}{H_2 g_1'} & \frac{1}{H_2}\left(\frac{1}{g_2'} + \frac{1}{g_1'}\right) & \frac{-1}{H_2 g_2'} & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \frac{-1}{H_N g_{N-1}'} & \frac{1}{H_N g_{N-1}'} \end{bmatrix} \tag{5.4}$$

while B differs from atmosphere to ocean,

$$ {}^{\mathsf{o}}\mathsf{B} = f_0 \begin{bmatrix} \frac{1}{H_1} & \frac{-1}{H_1} & & & & \\ & \frac{1}{H_2} & \frac{-1}{H_2} & & & \\ & & \cdot & \cdot & & \\ & & & \cdot & \cdot & \\ & & & & \frac{1}{H_N} & \frac{-1}{H_N} \end{bmatrix} \tag{5.5} $$

and ${}^{\mathsf{a}}\mathsf{B} = -{}^{\mathsf{o}}\mathsf{B}$. Note that B is a $N \times N+1$ matrix which produces two forcing terms on each layer, one at the top and one at the bottom. The entrainment vectors are given by

$$ {}^{\mathbf{o}}\mathbf{e} = \begin{bmatrix} {}^{o}w_{ek} \\ -\frac{\Delta_m^o T}{2\Delta_1^o T}\, {}^{o}w_{ek} \\ 0 \\ \cdot \\ \cdot \\ \frac{\delta_{ek}}{2f_o}\nabla_H^2\, {}^{o}p_N \end{bmatrix} \tag{5.6} $$

and

$$ {}^{\mathbf{a}}\mathbf{e} = \begin{bmatrix} {}^{a}w_{ek} \\ \frac{F_m^{\uparrow}+F_1^{\downarrow}-F_N^{\uparrow}}{{}^{a}\rho^{a}C_p\Delta_1^{a}T} \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}. \tag{5.7} $$

The entrainment equations will be altered if convection occurs. The dynamic topography vectors are simply

$$ {}^{\mathbf{a}}\tilde{\mathbf{D}} = \begin{bmatrix} \frac{f_0\, {}^{a}D}{{}^{a}H_1} \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \tag{5.8} $$

and

$$ {}^{\mathbf{o}}\tilde{\mathbf{D}} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \frac{f_0\, {}^{o}D}{{}^{o}H_N} \end{bmatrix}. \tag{5.9} $$

5.2 Ocean mixed layer

From (3.21),

$$ {}^{o}T_{mt} + ({}^{o}u_m\, {}^{o}T_m)_x + ({}^{o}v_m\, {}^{o}T_m)_y - \frac{{}^{o}w_{ek}\, {}^{o}T_m}{{}^{o}H_m} = {}^{o}K_2\nabla_H^2\, {}^{o}T_m - {}^{o}K_4\nabla_H^4\, {}^{o}T_m + \frac{{}^{o}F_0 + {}^{o}F_m^{e+}}{{}^{o}\rho\, {}^{o}C_p\, {}^{o}H_m}, \tag{5.10} $$

where total oceanic mixed layer forcing is given by

$$ {}^{o}F_0 = -F_\lambda - F_0^{\uparrow} - F_m^{\downarrow} - F_s, \tag{5.11} $$

and we have retained the entrainment flux as a separate term in (5.10). Note that the advective terms here use the mixed layer velocity which is modified by stress, rather than the geostrophic velocity used in (5.1).

5.3 Atmosphere mixed layer

From (3.18) we get

$$ {}^a h_{mt} + ({}^a u_m {}^a h_m)_x + ({}^a v_m {}^a h_m)_y = -{}^a e_m. \tag{5.12} $$

Note that ${}^a h_m$ is constrained to prevent very small values occurring: when ${}^a h_m$ becomes smaller than a critical value, it is reset to the critical value, and ${}^a T_m$ adjusted accordingly to conserve heat. Equation (2.3) gives

$$ {}^a T_{mt} + ({}^a u_m {}^a T_m)_x + ({}^a v_m {}^a T_m)_y + \frac{{}^a w_{ek} {}^a T_m}{{}^a H_m} = {}^a K_2 \nabla_H^2 {}^a T_m - {}^a K_4 \nabla_H^4 {}^a T_m + \frac{-{}^a F_m + {}^a F_0}{{}^a \rho {}^a C_p {}^a h_m}, \tag{5.13} $$

where atmospheric mixed layer forcing above and below the mixed layer are respectively given by

$$ {}^a F_m = F_m^{\uparrow} + F_1^{\downarrow} + {}^a F_m^{e-} \tag{5.14} $$

$$ {}^a F_0 = \begin{cases} F_m^{\downarrow} + F_\lambda + F_0^{\uparrow} & \text{over ocean,} \\ -F_s' & \text{over land.} \end{cases} \tag{5.15} $$

5.4 Convection

Convection will occur when unstable stratification forms and will act to change temperature in the mixed layers in a way which conserves heat. In the atmosphere, when ${}^a T_m$ exceeds ${}^a T_1$ we adjust the layer 1 entrainment as follows,

$$ \delta^a e_1 = \frac{{}^a h_m ({}^a T_1 - {}^a T_m)}{2 \Delta^a t \Delta_1^a T}, \tag{5.16} $$

and set ${}^a T_m = {}^a T_1$. Likewise, in the ocean, when ${}^o T_m < {}^o T_1$ we alter layer 1 entrainment by

$$ \delta^o e_1 = \frac{-{}^o H_m ({}^o T_1 - {}^o T_m)}{2 \Delta^o t \Delta_1^o T}, \tag{5.17} $$

and set ${}^o T_m = {}^o T_1$.

6 Mean state

The mean state is defined by the solution of the above equations when $F_s = \overline{F_s}$. In this state there will be no motion, so that pressure gradients, vorticity gradients and entrainment disappear. Our full equation set then reduces to a radiation balance between the atmosphere and the oceanic mixed layer. From (5.7) we get a heat balance in the upper atmosphere

$$ \overline{{}^a F_m^{e+}} + \overline{F_1^{\downarrow}} + \overline{F_m^{\uparrow}} - \overline{F_N^{\uparrow}} = 0, \tag{6.1} $$

(5.13) gives us the heat balance in the atmospheric mixed layer

$$ -\overline{F_m^{\uparrow}} - \overline{F_1^{\downarrow}} - \overline{{}^a F_m^{e-}} + [\overline{F_m^{\downarrow}} + \overline{F_\lambda} + \overline{F_0^{\uparrow}}] = 0, \tag{6.2} $$

and (5.10) can be used for the heat balance in the ocean mixed layer,

$$ -\overline{F_\lambda} - \overline{F_0^{\uparrow}} - \overline{F_m^{\downarrow}} - \overline{F_s} + \overline{{}^o F_m^{e+}} = 0, \tag{6.3} $$

where ${}^o F_m^{e+} = 0$ and ${}^a e_m = 0$. These equations can be manipulated to write

$$ \overline{F_N^{\uparrow}} = -\overline{F_s}, \tag{6.4} $$

which yields $\overline{{}^a T_m}$ provided one has knowledge of temperature, transmissivity and layer height in the atmosphere. Equation (6.3) can be written as a quartic equation in $\overline{{}^o T_m}$ to give oceanic mixed layer temperature. The value of ${}^a \phi_m$ is prescribed, and export heat fluxes are adjusted to satisfy (6.2).

Parameters	Value	Description
$({}^aX, {}^aY)$	(15360, 7680) km	Domain size
${}^aH_k$	(2000, 3000, 4000) m	Layer heights
${}^aH_m$	1000 m	Mean mixed layer height
${}^aH_{mmin}$	100 m	Minimum mixed layer height
${}^aT_k$	(330, 340, 350) K	Potential temperature structure
$\Delta^a x$	120 km	Horizontal grid spacing
$\Delta^a t$	5 min	Timestep
${}^a\rho$	1 kg/m^3	Density
${}^aC_p$	1000 J/kg/K	Specific heat capacity
${}^ag'_k$	(1.2, 0.4) m/s^2	Reduced gravity
C_D	1.3×10^{-3}	Drag coefficient
${}^aA_4$	2×10^{14} m^4/s	Horizontal viscosity coefficient
${}^a\alpha_{bc}$	1	Mixed boundary condition coefficient
${}^aK_2$	2.7×10^4 m^2/s	Horizontal temperature diffusion coefficient
${}^aK_4$	3×10^{14} m^4/s	Horizontal 4th order diffusion coefficient
K_η	2×10^5 m^2/s	Horizontal diffusion coefficient for ${}^a\eta_m$
${}^a\phi_m$	0.15 W/m^3	Relaxation rate for mixed layer height
σ	5.6704×10^{-8} W/m^2/K^4	Stefan–Boltzmann constant
γ	0.01 K/m	Adiabatic lapse rate
$\overline{F_s}$	-210 W/m^2	Mean incoming radiation
$\lvert F'_s \rvert$	90 W/m^2	Amplitude of variable incoming radiation
λ	35 W/m^2/K	Sensible and latent heat flux coefficient
z_m	200 m	Optical depth in the mixed layer
z_k	(30 000, 30 000, 40 000) m	Optical depths

Table 1: Specified default atmospheric parameters for Q-GCM.

7 Final equation set

The full set of equations for the two layer version is now written in the order in which they are solved in the model. The equation numbers used here are referred to in comments in the code. Specified parameters are shown in Tables 1 and 2.

At the start of each ocean timestep we find the atmospheric stress from (3.3), (3.5a) and (3.5b):

$$({}^a\tau^x, {}^a\tau^y) = \frac{C_D|\mathbf{^a u_m}|}{\chi}\left({}^au_1 - \frac{C_D|\mathbf{^a u_m}|}{{}^aH_m f_0}{}^av_1, {}^av_1 - \frac{C_D|\mathbf{^a u_m}|}{{}^aH_m f_0}{}^au_1\right), \tag{7.1}$$

where

$$({}^au_1, {}^av_1) = \frac{1}{f_0}\left(-{}^ap_{1y}, {}^ap_{1x}\right), \tag{7.2}$$

$$|\mathbf{^a u_m}| = \frac{{}^aH_m f_0}{\sqrt{2}\,C_D}\sqrt{-1 + \sqrt{1 + 4\left(\frac{C_D|\mathbf{^a u_1}|}{{}^aH_m f_0}\right)^2}}. \tag{7.3}$$

and

$$\chi \equiv 1 + \left(\frac{C_D|\mathbf{^a u_m}|}{{}^aH_m f_0}\right)^2. \tag{7.4}$$

We use this to find the ocean stress from (3.9)

$${}^o\tau = \xi\frac{{}^a\rho}{{}^o\rho}{}^a\tau, \tag{7.5}$$

Parameters	Value	Description
$({}^oX, {}^oY)$	(3840, 4800) km	Domain size
${}^oH_k$	(300, 1100, 2600) m	Layer heights
${}^oH_m$	100 m	Mixed layer height (fixed)
${}^oT_k$	(278, 268, 258) K	Potential temperature structure
$\Delta^o x$	10 km	Horizontal grid spacing
$\Delta^o t$	30 min	Timestep
${}^o\rho$	1000 kg/m^3	Density
${}^oC_p$	4000 J/kg/K	Specific heat capacity
${}^og'_k$	(0.05, 0.025) m/s^2	Reduced gravity
δ_{ek}	1 m	Bottom Ekman layer thickness
${}^oA_2$	0 m^2/s	Laplacian horizontal viscosity coefficient
${}^oA_4$	2.0×10^{10} m^4/s	Biharmonic horizontal viscosity coefficient
${}^o\alpha_{bc}$	0.5	Mixed boundary condition coefficient
${}^oK_2$	380 m^2/s	Horizontal temperature diffusion coefficient
${}^oK_4$	4×10^{10} m^4/s	Horizontal 4th order diffusion coefficient
f_0	1×10^{-4} s^{-1}	Mean Coriolis parameter
β	2×10^{-11} (ms)$^{-1}$	Coriolis parameter gradient
ξ	1.0	Atmosphere–ocean stress multiplier

Table 2: Specified default oceanic parameters for Q-GCM.

which requires interpolating stress onto the oceanic pressure grid and multiplication by the constant $\xi^a\rho/{}^o\rho$. The Ekman velocities can then be calculated (naturally on the temperature grid, and then averaged onto the pressure grid) from (3.8) and (3.10)

$$ {}^aw_{ek} = \frac{1}{f_0}({}^a\tau^y_x - {}^a\tau^x_y), \tag{7.6} $$

$$ {}^ow_{ek} = \frac{1}{f_0}({}^o\tau^y_x - {}^o\tau^x_y). \tag{7.7} $$

The radiative fluxes at the atmospheric mixed layer boundaries are determined at this stage from (5.13)

$$ {}^aF_m = \sum_{i=1}^{N-1} A^{\downarrow}_{1,i}{}^a\eta_i + ({}^a\phi_m + B^{\uparrow}_m + B^{\downarrow}_1){}^a\eta_m + (C^{\uparrow}_m + C^{\downarrow}_1){}^aD + D^{\uparrow}_m{}^aT_m{}', \tag{7.8} $$

$$ {}^aF_0 = \begin{cases} D^{\downarrow}_m{}^aT_m{}' + D^{\uparrow}_0{}^oT_m{}' + \lambda({}^oT_m{}' - {}^aT_m{}') & \text{over ocean,} \\ -F'_s & \text{over land.} \end{cases} \tag{7.9} $$

In the ocean,

$$ {}^oF_0 = D^{\downarrow}_m{}^aT_m{}' + D^{\uparrow}_0{}^oT_m{}' + \lambda({}^oT_m{}' - {}^aT_m{}') + F'_s. \tag{7.10} $$

The next stage is to determine the evolution of the oceanic mixed layer using (5.10),

$$ {}^oT_{mt}{}' = -({}^ou_m{}^oT_m{}')_x - ({}^ov_m{}^oT_m{}')_y + {}^oK_2\nabla^2_H{}^oT_m{}' - {}^oK_4\nabla^4_H{}^oT_m{}' + \frac{{}^ow_{ek}({}^oT_1 + {}^oT_m{}')}{2{}^oH_m} - \frac{{}^oF_0}{{}^o\rho{}^oC_p{}^oH_m}. \tag{7.11} $$

This involves integrating forward in time. Note here that oceanic layer temperatures are defined relative to $\overline{{}^oT_m}$ so that terms including the mean temperature disappear. The entrainment at the interface between layers 1 and 2 is written as,

$$ {}^oe_1 = -\frac{\Delta^o_m T}{2\Delta^o_1 T}{}^ow_{ek} \tag{7.12} $$

After timestepping (7.11), if ${}^oT_m < {}^oT_1$, convection will occur. This requires a correction to the interfacial entrainment which is handled by adding

$$\delta^o e_1 = \frac{-{}^oH_m({}^oT_1 - {}^oT_m)}{2\Delta^o t \Delta_1^o T} \tag{7.13}$$

to the existing entrainment ${}^oe_1$. We then correct mixed layer temperature by setting ${}^oT_m = {}^oT_1$.

Next we are in a position to step the QG equations in all parts of the domain except for the boundaries using (5.2)

$$\mathbf{q}_t = \frac{1}{f_0} J(\mathbf{q}, \mathbf{p}) + \frac{A_2}{f_0}\nabla_H^4 \mathbf{p} - \frac{A_4}{f_0}\nabla_H^6 \mathbf{p} + \mathsf{Be}, \tag{7.14}$$

Note that the vorticity equation does not apply on boundary points, and the boundary conditions we use are formulated in terms of pressure. Pressure is determined by writing the vorticities in modal form. The modes are determined by numerically finding the eigenvalues of (5.3),

$$f_0\mathbf{q} - f_0\beta y = \nabla_H^2 \mathbf{p} - f_0^2 \mathsf{A}\mathbf{p} + f_0\tilde{\mathbf{D}}. \tag{7.15}$$

The techniques and routines used for the modal decomposition are described in §8.3. The decomposition allows direct solution for modal pressure from modal vorticity using a Helmholtz solver (see §8.4 for details). It is at this time that the ocean mass constraints are included (see Appendix B.1). Layer pressures are recovered from modal pressures. Boundary values of vorticity are recovered from the pressure field using mixed boundary conditions (see Appendix C).

We are then ready to timestep the atmosphere internal points. As in the ocean case we begin by calculating (7.6) on the temperature grid, and from (5.12) and (5.13) we get

$${}^a h_{mt} = -({}^a u_m {}^a h_m)_x - ({}^a v_m {}^a h_m)_y + K_\eta \nabla_H^2 {}^a h_m - \frac{{}^a\phi_m {}^a\eta_m}{{}^a\rho^a C_p \Delta_m^a T}, \tag{7.16}$$

where an extra diffusive term has been added for numerical stability, and mixed layer thickness is constrained not to fall below a specified minimum value. The temperature evolution is given by

$$\begin{aligned} {}^a T_{mt}{}' = -({}^a u_m {}^a T_m{}')_x - ({}^a v_m {}^a T_m{}')_y + {}^a K_2 \nabla_H^2 {}^a T_m{}' - \\ {}^a K_4 \nabla_H^4 {}^a T_m{}' - \frac{{}^a w_{ek} {}^a T_m{}'}{{}^a H_m} + \frac{-{}^a F_m + {}^a F_0}{{}^a\rho^a C_p {}^a h_m}. \end{aligned} \tag{7.17}$$

Entrainment at the top of layer 1 in the atmosphere comprises the entire radiation scheme in a single equation:

$$\begin{aligned} {}^a e_1 = \frac{1}{{}^a\rho^a C_p \Delta_1^a T}\Bigg[\sum_{i=1}^{N-1}\left(A_{1,i}^{\downarrow} - A_{N,i}^{\uparrow}\right){}^a\eta_i + \left(B_m^{\uparrow} + B_1^{\downarrow} - B_N^{\uparrow}\right){}^a\eta_m + \\ \left(C_m^{\uparrow} + C_1^{\downarrow} - C_N^{\uparrow}\right){}^a D + \left(D_m^{\uparrow} - D_N^{\uparrow}\right){}^a T_m{}'\Bigg]. \end{aligned} \tag{7.18}$$

Convection will occur if, after timestepping, ${}^a T_m{}' > {}^a T_1$, in which case we adjust the entrainment at the top of layer 1 by adding

$$\delta^a e_1 = \frac{{}^a h_m({}^a T_1 - {}^a T_m)}{2\Delta^a t \Delta_1^a T} \tag{7.19}$$

and then set the temperature to be ${}^a T_m{}' = {}^a T_1$.

Finally we step the atmosphere QG equations (7.14). As before, the pressures are found by writing the vorticity in modal form (see §8.3), using a Helmholtz solver, and then applying the momentum conditions described in Appendix B.2. Layer pressures are recovered from modal pressures, and boundary values of vorticity found from the pressure field.

8 Numerical formulation

Having derived the continuous equations which are the basis of the model, we must now formulate finite approximations to them in order to produce a computable solution. This chapter gives a brief account of the numerical methods used.

8.1 Discretisation in space and time

The domains are constructed so that (in the current configuration) there are 12 ocean gridlengths within each atmosphere gridlength, and the ocean occupies an integral number of atmosphere gridlengths (see figure 2). The other major difference between the atmosphere and ocean is that atmospheric velocities are much higher, and therefore, despite the larger grid cells, the atmosphere requires a smaller timestep. In this version of the model there are 6 atmosphere time steps within each ocean timestep. To conserve heat and momentum, forcing terms cannot be re-computed every atmosphere timestep: instead these terms are computed before each ocean timestep, and are held constant during the 6 atmosphere timesteps.

Both atmosphere and ocean variables are discretised horizontally on an Arakawa C-grid (Arakawa and Lamb, 1977) as shown in figure 2. The grid is Cartesian, with the same resolution in both x- and y-directions. Pressure and vorticity are tabulated at the same points (referred to consistently as p points), and temperature and mixed layer thickness are tabulated at the temperature (T) points. These are the prognostic fields; other quantities can be diagnosed from them at the appropriate gridpoints as required. When it is necessary to interpolate quantities from the coarse atmospheric grid to the finer oceanic grid, we use a simple and efficient bilinear interpolation scheme which is conservative.

The advective terms in mixed layer evolution equations such as (7.11), (7.16) and (7.17) are formulated in the usual manner for C-grid models, in which the advective velocities are computed at the centres of the faces of the cell containing a T gridpoint. This ensures that the advection scheme is conservative, and that unlike some more sophisticated advection schemes there is no implicit diffusion; it is all explicit. The quantity being advected is approximated by the average of its values on either side of the face.

The exception to this is the advection of vorticity in equations (7.14). Here we use a higher order (9-point) Arakawa Jacobian (Arakawa and Lamb, 1981) with superior conservation properties: it conserves both energy and enstrophy.

The timestepping of the evolution equations uses a leapfrog scheme, which is second order in time. Occasional averaging of the two time levels carried by the program is performed, to suppress the computational mode which is inevitable with this scheme.

8.2 Effect of the Laplacian operator on a Fourier component

This section considers the effect of the $\frac{\partial^2}{\partial x^2}$ operator on a single Fourier component; the results will prove useful in a couple of later sections. Consider a one-dimensional Fourier expansion (the extension to two dimensions is straightforward):

$$p(x) = \sum_k \tilde{p}_k e^{2\pi i k x}. \tag{8.1}$$

In the continuous case, applying the second derivative operator gives

$$\frac{d^2 p}{dx^2} = \sum_k -\tilde{p}_k (2\pi k)^2 e^{2\pi i k x}. \tag{8.2}$$

In the finite difference case, $\frac{d^2 p}{dx^2}$ at x is given by the centred-difference approximation

$$\frac{d^2 p}{dx^2} = \frac{p(x+dx) - 2p(x) + p(x-dx)}{dx^2}. \tag{8.3}$$

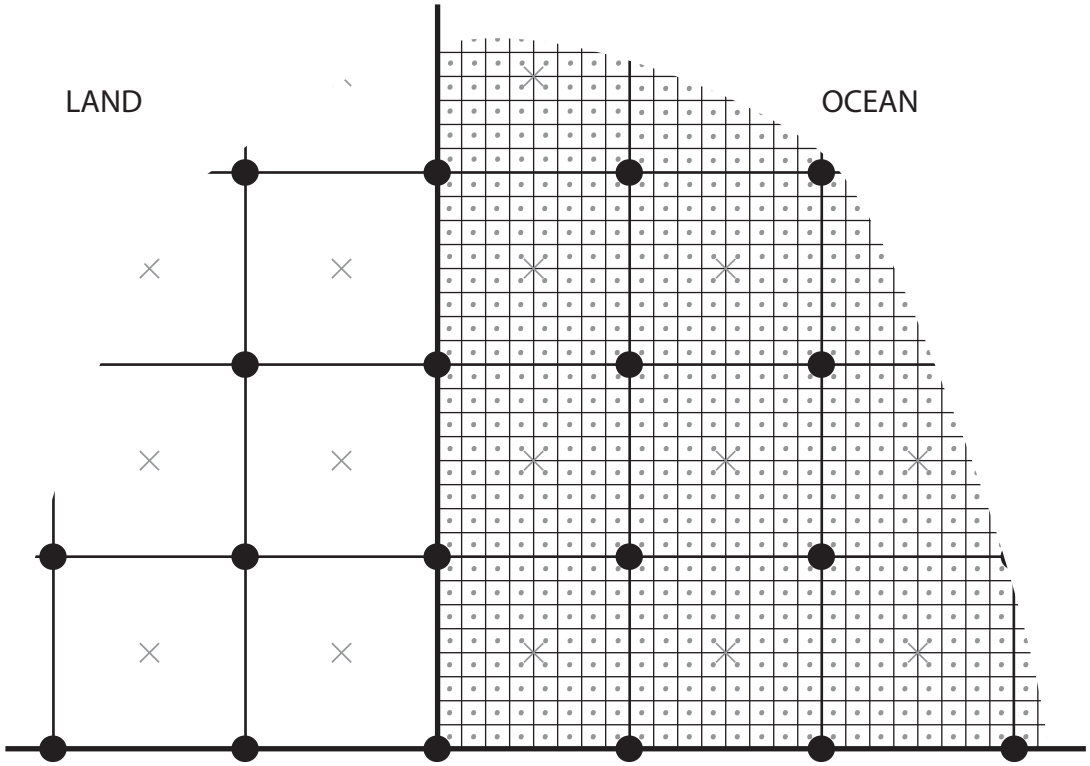

Figure 2: Cutaway section showing the grid used in the implementation of the model near the land-ocean boundary. The atmosphere grid is the larger grid, with • showing the pressure grid points and × showing the temperature gridpoints. The ocean grid is also shown, with eight ocean gridlengths to every atmosphere gridlength. The ocean pressure gridpoints are at the vertices of the grid shown, and ocean temperature points indicated by ·.

where dx is the gridspacing. Substituting (8.1) and simplifying somewhat gives

$$\frac{d^2p}{dx^2} = \sum_k \tilde{p}_k \left(\frac{e^{2\pi ikdx} + e^{-2\pi ikdx} - 2}{dx^2} \right) e^{2\pi ikx}, \tag{8.4}$$

i.e.

$$\frac{d^2p}{dx^2} = \sum_k \tilde{p}_k \left(\frac{2cos(2\pi kdx) - 2}{dx^2} \right) e^{2\pi ikx}, \tag{8.5}$$

which, using some familiar trigonometric identities, reduces to:

$$\frac{d^2p}{dx^2} = \sum_k -\tilde{p}_k \left(\frac{2sin(\pi kdx)}{dx} \right)^2 e^{2\pi ikx}. \tag{8.6}$$

So the finite difference result (8.6) is similar to the continuous result (8.2), but with the replacement

$$\pi k \Rightarrow \frac{sin(\pi kdx)}{dx}, \tag{8.7}$$

in the Fourier coefficient. For small k (well-resolved wavelengths) these are equivalent.

8.3 Numerical decomposition into modes

For both ocean and atmosphere the relationship between pressure and vorticity can be written in matrix form as in (5.3), where A is the tridiagonal matrix given by (5.4). To decompose the layer quantities into modes we suppose that A has left eigenvectors $\mathbf{L}_m$, right eigenvectors $\mathbf{R}_m$ and eigenvalues λ_m , where m is the eigenmode number, ranging from 1 to N. Then

$$\mathsf{A}\mathbf{R}_m = \lambda_m \mathbf{R}_m$$
$$\mathbf{L}_m^T \mathsf{A} = \lambda_m \mathbf{L}_m^T$$

for each mode m. It can be shown that $\mathbf{L}_m$ and $\mathbf{R}_m$ are orthogonal, i.e $\mathbf{L}_m^T \cdot \mathbf{R}_n = 0$ unless $m = n$. Note that since A is nonsymmetric $\mathbf{L}_m \neq \mathbf{R}_m$ in general.

From the particular structure of A, it follows that the eigenvalues will all be real, and for the sign of A chosen in (5.3) $\lambda_m \geq 0$. The dimensions of λ_m are $1/c_m^2$, where c_m is a phasespeed in m s^{-1}. One of the eigenmodes will have $\lambda_m = 0$; we call this the barotropic mode. We order the remaining $N-1$ modes with λ_m increasing (c_m decreasing); these are the baroclinic modes.

The eigenvalues and eigenvectors of A can be computed using standard linear algebra methods (see e.g. Golub and van Loan, 1983), and this is done in the subroutine `eigmod`. We choose to use routines from the package `LAPACK` for portability. We can then use the eigenmodes of A to separate the set of equations (7.15) into an easily soluble set, as follows:

Expand the pressure vector $\mathbf{p}$ in terms of the right eigenvectors $\mathbf{R}_n$

$$\mathbf{p} = \sum_n \hat{p}_n \mathbf{R}_n. \tag{8.8}$$

Premultiply (8.8) by $\mathbf{L}_m^T$, then

$$\mathbf{L}_m^T \cdot \mathbf{p} = \sum_n \hat{p}_n \mathbf{L}_m^T \mathbf{R}_n = \hat{p}_m \mathbf{L}_m^T \cdot \mathbf{R}_m, \tag{8.9}$$

by orthogonality, so that

$$\hat{p}_m = \mathsf{C}_{\mathrm{l2m}} \mathbf{p}, \tag{8.10}$$

where $\mathsf{C}_{\mathrm{l2m}}(m,k) = \frac{\mathbf{L}_m^T(k)}{\mathbf{L}_m^T \cdot \mathbf{R}_m}$ is an $N \times N$ matrix used to convert from layer to modal amplitudes. Note that for reasons of runtime efficiency the program carries the transpose of $\mathsf{C}_{\mathrm{l2m}}$.

We also require a matrix $\mathsf{C}_{\mathrm{m2l}}$ for converting from modal to layer amplitudes. Clearly from (8.8) $\mathsf{C}_{\mathrm{m2l}}(k,m) = \mathbf{R}_m(k)$. It follows that the product of the matrices $\mathsf{C}_{\mathrm{m2l}}$ and $\mathsf{C}_{\mathrm{l2m}}$ is the identity matrix, as required for consistency. This can be confirmed (to machine precision) by subroutine `eigmod`.

Substitute the expansion (8.8) into the matrix equation (7.15):

$$f_0(\mathbf{q} - \tilde{\mathbf{D}} - \beta y) = \nabla_H^2 \sum_m \hat{p}_m \mathbf{R}_m - f_0^2 \sum_m \hat{p}_m \lambda_m \mathbf{R}_m, \tag{8.11}$$

where the quantity $f_0^2 \lambda_m$ has dimensions $1/{r_m}^2$, where r_m is a deformation radius (which is infinite for the barotropic mode). So we have

$$f_0(\mathbf{q} - \tilde{\mathbf{D}} - \beta y) = \sum_m \left(\nabla_H^2 - \frac{1}{{r_m}^2} \right) \hat{p}_m \mathbf{R}_m. \tag{8.12}$$

Premultiplying by $\mathbf{L}_k^T$, summing over the layers (k), and dividing by the dot product, we obtain the equation for the amplitude of each mode:

$$\left(\nabla_H^2 - \frac{1}{{r_m}^2} \right) \hat{p}_m = f_0 \sum_k \mathsf{C}_{\mathrm{l2m}}(m,k)(q_k - \tilde{D}_k - \beta y). \tag{8.13}$$

So we need to solve a two-dimensional modified Helmholtz equation with a known RHS for each modal amplitude $\hat{p}_m$, using the method given in section 8.4, and convert these to layer amplitudes using (8.8).

8.4 Solution of the modified Helmholtz equation

Most of the equations which constitute the model define one variable explicitly in terms of other variables or their derivatives. The exceptions are the modified Helmholtz equations (8.13) which define pressures *implicitly* in terms of vorticities, subject to suitable boundary conditions. We require a computationally efficient solution method for such equations. Write the general inhomogeneous Helmholtz equation as

$$\left(\nabla_H^2 - \frac{1}{{r_m}^2}\right) p = R \tag{8.14}$$

where the right hand side $R(x, y)$ is a known function and r_m is the Rossby radius for a given mode. The barotropic mode is a special case for which r_m is infinite, and (8.14) reduces to Poisson's equation.

We will consider the solution to (8.14) in a channel periodic in x (the atmospheric case). The method carries over to the finite ocean box with simple modifications to be discussed later. Assume the pressure $p(x, y)$ and the right hand side $R(x, y)$ are tabulated on a regular grid at positions $x_i = (i-1)dx$ (M points), $y_j = y_s + (j-1)dy$ (N points), so that R is an $M \times N$ matrix R_{ij}. Expand p and R as one-dimensional Fourier series in the x-direction at each of the $(N-2)$ values y_j within the domain (i.e. not including the zonal boundaries $j = 1$ and $j = N$):

$$p_{ij} = \sum_k \tilde{p}_{k,j} e^{2\pi i k x_i}. \tag{8.15}$$

We can use a Fast Fourier Transform (FFT) routine to efficiently compute the array of coefficients $\tilde{R}_{k,j}$ from the known R_{ij}. The particular ordering of the values of k in the transform vector depends on the FFT package used, so we ignore such details in this general account.

Then from (8.6)

$$\frac{\partial^2 p_{ij}}{\partial x^2} = \sum_k -\tilde{p}_{k,j} \left(\frac{2sin(\pi k dx)}{dx}\right)^2 e^{2\pi i k x_i}. \tag{8.16}$$

$\frac{\partial^2 p_{ij}}{\partial y^2}$ at $y = y_j$ is given by the usual centred difference approximation (8.3)

$$\frac{d^2 p_{ij}}{dy^2} = \sum_k \left(\frac{\tilde{p}_{k,j+1} - 2\tilde{p}_{k,j} + \tilde{p}_{k,j-1}}{dy^2}\right) e^{2\pi i k x_i} \tag{8.17}$$

so at each internal point i, j we have an equation of the form

$$\sum_k -\tilde{p}_{k,j} \left(\left(\frac{2sin(\pi k dx)}{dx}\right)^2 + \frac{1}{{r_m}^2}\right) e^{2\pi i k x_i} + \sum_k \left(\frac{\tilde{p}_{k,j+1} - 2\tilde{p}_{k,j} + \tilde{p}_{k,j-1}}{dy^2}\right) e^{2\pi i k x_i} = \sum_k \tilde{R}_{k,j} e^{2\pi i k x_i}. \tag{8.18}$$

But for this to be true for at all the points $x = x_i$, it must be true for each Fourier component $e^{2\pi i k x}$ separately, i.e. for all k and j, we must have

$$-\tilde{p}_{k,j} \left(\left(\frac{2sin(\pi k dx)}{dx}\right)^2 + \frac{1}{{r_m}^2}\right) + \frac{(\tilde{p}_{k,j+1} - 2\tilde{p}_{k,j} + \tilde{p}_{k,j-1})}{dy^2} = \tilde{R}_{k,j}. \tag{8.19}$$

For each value of the M values of k, we can thus write the set of $(N-2)$ equations (8.19) for the range of internal j values as a single tridiagonal matrix equation of size $(N-2) \times (N-2)$

relating the unknown vector $\tilde{p}_{k,j}$ and the known vector $\tilde{R}_{k,j}$:

$$\begin{bmatrix}
\cdot & \cdot & & & & & & & & & \\
\cdot & \cdot & \cdot & & & & & & & & \\
 & \cdot & \cdot & \cdot & & & & & & & \\
 & & \cdot & \cdot & \cdot & & & & & & \\
 & & & & & & & & & & \\
 & & & a & b_k & a & & & & & \\
 & & & & a & b_k & a & & & & \\
 & & & & & a & b_k & a & & & \\
 & & & & & & & & & & \\
 & & & & & & & \cdot & \cdot & \cdot & \\
 & & & & & & & & \cdot & \cdot & \cdot \\
 & & & & & & & & & \cdot & \cdot \\
 & & & & & & & & & &
\end{bmatrix} \tilde{p}_{k,j} = \tilde{R}_{k,j} \tag{8.20}$$

where dots denote non-zero coefficients, and the a and b coefficients are known:

$$a = \frac{1}{dy^2}, \tag{8.21}$$

$$b_k = -\left(\left(\frac{2sin(\pi k dx)}{dx}\right)^2 + \frac{1}{{r_m}^2}\right) - 2a \tag{8.22}$$

Note that to maintain strictly tridiagonal form, only two entries appear on the top and bottom rows of the matrix. This is equivalent to asserting that $\tilde{p}$ vanishes on the zonal boundaries $j = 1$ and $j = N$ for all k, and thus $p = 0$ also.

Such a tridiagonal system can be efficiently solved for the unknown vector without a full matrix inversion; see any good numerical linear algebra textbook, such as Press et al. (1992). For the modified Helmholtz operator, it is clear from (8.22) that $|b_k| \geq 2|a|$ for all k, so the matrix is "diagonally dominant", and no problems of zero pivots can occur. The tridiagonal solution method is built into the subroutines `chsolv` and `solb`. Having solved the system for $\tilde{p}_{k,j}$ at all M wavenumbers k, we can then use (8.15) at each latitude to get the pressure p_{ij} at all gridpoints. Thus we have solved the inhomogeneous Helmholtz equation (8.14) in a zonally periodic channel, subject to the condition $p = 0$ on the zonal boundaries.

For the ocean, where we require p to be constant around the entire boundary, we use Fourier sine transforms instead of the periodic Fourier transforms, to ensure that $p = 0$ on the meridional boundaries too.

Note that the various Fourier transform routines use a significant fraction of the total CPU time (16%–18% for the default case, depending on the computer system used). It is therefore important that the parameters determining the transform lengths (`nxa`, `ndxr` and `nxaooc`, all set in `parameter.src`) should be chosen to have numerous small prime factors, so as to maximise the efficiency of the transform routines.

We can also use this scheme to solve the homogeneous Helmholtz equation for baroclinic modes. The homogeneous equation is:

$$\left(\nabla_H^2 - \frac{1}{{r_m}^2}\right) p = 0. \tag{8.23}$$

Write $p = \mathcal{L} + s$, where $\mathcal{L}$ is a known solution of Laplace's equation $\nabla_H^2 \mathcal{L} = 0$ which is non-zero on the boundaries; then s satisfies

$$\left(\nabla_H^2 - \frac{1}{{r_m}^2}\right) s = \frac{\mathcal{L}}{{r_m}^2}, \tag{8.24}$$

which is of the same form as (8.14). s satisfies the usual condition $s = 0$ on the solid boundaries of the domain, so $p = \mathcal{L}$ on these boundaries. We can add any multiple of these homogeneous solutions to the inhomogeneous solution to obtain any required value of p on the solid boundaries.

8.5 Integration routines

Several of the radiative flux coefficients introduced in section 8.2 involve vertical integrals over the appropriate layers. A subroutine `trapin` is provided which computes the extended trapezoidal rule approximation to the integral of a tabulated function (Press et al., 1992). Two routines, `xintt` and `xintp`, are provided for computing area integrals. `xintt` computes the area integral of a quantity tabulated at T points, as a simple sum. `xintp` computes the area integral of a quantity tabulated at p points, as a simple sum over interior values, with an additional factor controlling the contribution of edge values to the integral.

8.6 Diffusive timescales

Several of the equations contain diffusive terms of second or fourth order (recall that the ∇_H^6 term in (7.14) was originally introduced as fourth order terms in (2.7a,b), and is effectively fourth order since $q \sim \nabla_H^2 p$). To understand the damping effect of the diffusion coefficients, it is useful to derive their associated timescales.

Consider some field $p(x, y)$, and introduce second and fourth order diffusive terms into its evolution equation, with coefficients A_2 and A_4 respectively, so that

$$\frac{\partial p}{\partial t} = \ldots + A_2 \nabla_H^2 p - A_4 \nabla_H^4 p. \tag{8.25}$$

We will require $A_2, A_4 > 0$ to ensure decay. Expand p as a two-dimensional Fourier transform:

$$p(x, y) = \sum_k \sum_l \tilde{p}_{kl} e^{2\pi i(kx+ly)} \tag{8.26}$$

then, differentiating the Fourier expansion, we find that each term will cause exponential decay of the coefficient $\tilde{p}_{kl}$, with timescales τ_2, τ_4 given in the finite difference case (using (8.6)) by

$$\frac{1}{\tau_2} = A_2 \left(\left(\frac{2sin(\pi k\delta)}{\delta} \right)^2 + \left(\frac{2sin(\pi l\delta)}{\delta} \right)^2 \right) \tag{8.27}$$

$$\frac{1}{\tau_4} = A_4 \left(\left(\frac{2sin(\pi k\delta)}{\delta} \right)^2 + \left(\frac{2sin(\pi l\delta)}{\delta} \right)^2 \right)^2, \tag{8.28}$$

where δ is the (equal) gridlength in either direction. The main program evaluates these timescales for two special cases:

1. A circular eddy whose radius is the baroclinic Rossby radius r_m, so $k = l = \frac{1}{2r_m}$,

$$\frac{1}{\tau_2} = 2A_2 \left(\frac{2sin(\pi\delta/2r_m)}{\delta} \right)^2 \tag{8.29}$$

$$\frac{1}{\tau_4} = 4A_4 \left(\frac{2sin(\pi\delta/2r_m)}{\delta} \right)^4 \tag{8.30}$$

2. A one-dimensional wave at the highest wavenumber representable on the grid (two grid-point noise), $k = \frac{1}{2\delta}$, $l = 0$

$$\frac{1}{\tau_2} = A_2 \left(\frac{2}{\delta} \right)^2 \tag{8.31}$$

$$\frac{1}{\tau_4} = A_4 \left(\frac{2}{\delta} \right)^4 \tag{8.32}$$

9 Users' guide

Having briefly described the numerical formulation of the model, we now discuss the practicalities of using the code. The code is available in gzipped tar format, which expands to a single directory.

9.1 Components of the code

The code for Q-GCM consists of a number of Fortran77 source files (`*.f/*.F`), various common block files (`*.cmn`), a file called `parameter.src` and a generic `Makefile`. The code uses three public domain software packages:

FFTPACK This is a free, portable library of Fortran routines for computing 1-D fast Fourier transforms, originally written by Paul N. Swarztrauber of NCAR, Boulder. There is an FFTPACK webpage at NCAR:
`http://www.scd.ucar.edu/softlib/FFTPACK.html`
but the package is also distributed as part of the Netlib software repository:
`http://www.netlib.org/`
in Tennessee, USA, and its various mirror sites, which include:
`http://www.mirror.ac.uk/sites/netlib.bell-labs.com/netlib/` (UK)
`http://netlib.uow.edu.au/` (Australia).

We use the double precision version of the library, which can usually be found in the subdirectory `bihar`. Alternatively we also distribute our own versions of the necessary routines, modified to remove some programming constructs such as "computed GOTOs" which are deprecated in Fortran90, and which cause compiler warnings (directory `newbihar`). The FFTPACK routines are included via the file `fftpack.f` which you will have to edit to point to the location of these files on your system.

LAPACK This is a free, public domain linear algebra package. It is included in the scientific software libraries provided with most commercial Fortran compilers, and we recommend you access it in this way. Otherwise the source code can be downloaded from Netlib and included in the same manner as FFTPACK. The file `lasubs.f` lists the LAPACK and underlying BLAS routines which are required.

netCDF available from
`http://www.unidata.ucar.edu/packages/netcdf/`
NetCDF (network Common Data Format) is a machine-independent, self-documenting format for representing scientific data, and a library of routines for reading and writing data in this format. You will need to have installed the netCDF libraries on your system, and to edit the `Makefile` to indicate where they are. A version of the include file `netcdf.inc` (edited to eliminate some compiler warnings) is included in the Q-CGM distribution. Our development and testing has used version 3.5.0 of netCDF.

There are several easy ways to access the data in netCDF format. One way to view the data is with the browser package ncview:
`http://www.gfdl.gov/~jps/GFDL_VG_2DPackages.html`.
We also use the netCDF toolbox for Matlab :
`http://woodshole.er.usgs.gov/staffpages/cdenham/public_html`.

It is our policy to use public domain packages where possible, so that anyone may use Q-GCM without having to pay for package licences.

9.2 Compiling the code

The `Makefile` will take care of the compiling and linking, but it contains preprocessor options and a few flags specific to your system which must be set before it will work:

atmos_only, ocean_only Flags which, if set, restrict the model to run in atmosphere/ocean only mode. At most one of these flags should be set.

get_covar Flag to determine whether the covariance matrices should be computed. If set, this option turns on routines which accumulate the corrected sums of squares and products, which enable the covariance matrix and hence time-independent EOFS and principal components to be found in post-processing. This option significantly increases the memory requirement of the program.

use_netcdf Flag to make use of the netCDF output facility (see below). Turn this flag off to run the model without netCDF output.

small_local Flag to activate Fortran SAVE statements in those subroutines with significant internal storage. Needed on systems which have a small kernel limit on the amount of temporary (stack) storage which can be allocated. Ideally this flag should not need to be set: try using it if your system (linux only?) gives a `Segmentation Fault` error message which is not fixed by setting environment stack size limits as described in section 9.3.8. This flag also turns on diagnostics indicating the approximate amount of stack storage allocated by those subroutines with significant local arrays.

The above items control the treatment of `*.F` files by the Fortran preprocessor.

FC The command which invokes your Fortran compiler (e.g. `f90`, `f77`, `ifc`, `pgf90`, ...)

FFLAGS List of flags for your Fortran compiler (to control optimization/debugging/profiling etc.). See your Fortran compiler's `man` pages for details.

LAPACK Compiler option to pick up the LAPACK library

NCDIR The top directory of your netCDF installation

NCLIB The flag showing where to find the library of your netCDF installation

IDIR Directory containing the netCDF include file `netcdf.inc`. This will be the current directory if you are using the version of this file included in the distribution.

Compilation has been tested on several systems, and the `Makefile` contains some example options (commented out) from each of these systems, along with information on the operating system and compiler used. Simply type `make q-gcm` to compile, once you have edited the `Makefile` to suit your system.

The model has been run using both `f77` and `f90/f95` compilers. The model can run in parallel; instructions using the standard OpenMP API are included in the code. You will need to set a suitable compiler flag to enable these directives. The model is explicitly parallelised. We do not recommend the use of an auto-paralleliser in addition to the OpenMP parallelisation in the code; this degrades the parallel performance on some systems.

9.3 Structure of the code

The Q-GCM model consists of a main program and 61 subprograms, 7 of which remain with the main program in the file `q-gcm.F`, the remainder having been separated out into other `*.f/*.F` files. Below is a list showing where to find subroutines in the `*.f/*.F` files:

`q-gcm.F`	contains `qcomp*`, `merqat*`, `zeroin*`, `rbalin*`, `prsamp*`, `resave`, `memreq` (7)
`amlsubs.f`	contains `aml*`, `amladf*` (2)
`areasubs.F`	contains `areavg`, `areint*` (2)
`atisubs.f`	contains `atinvq*`, `chsolv*` (2)
`conhoms.F`	contains `constr*`, `homsol*` (2)
`covsubs.F`	contains `covini*`, `covatm`, `covocn`, `covout`, `tsampl*` , `psambx*` , `psamch*`, `dssp*` (8)
`diasubs.F`	contains `diagno*`, `del4bx*`, `del4ch*`, `genint*` (4)
`intsubs.f`	contains `xintt*`, `xintp*` (2)
`nc_subs.F`	contains `handle_err`, `ocnc_init`, `atnc_init`, `ocnc_out*`, `atnc_out*` , `monnc_init`, `monnc_out`, `resave_nc`, `restart_nc` (9)
`ocisubs.f`	contains `ocinvq*`, `solb*` (2)
`omlsubs.f`	contains `oml*`, `omladf*` (2)
`qgasubs.f`	contains `qgastep*`, `atadif*` (2)
`qgosubs.f`	contains `qgostep*`, `ocadif*` (2)
`radsubs.f`	contains `radiat*`, `trapin*` (2)
`tavsubs.F`	contains `tavini*`, `tavatm*`, `tavocn*`, `tavout` (4)
`topsubs.F`	contains `topset`, `topout_nc` (2)
`xfosubs.F`	contains `xforc*`, `function fsprim`, `bilint*` (3)
	`atqzbd.f*`, `eigmod.f`, `ocqbdy.f*` and `valids.F*` are single routine files (4)

In addition there are linear algebra library routines from LAPACK and Fourier transform library routines from FFTPACK (see `fftpack.f`) not counted above. Two additional code fragments `in_param.f` and `out_param.f` are also added to the main program by `INCLUDE` statements. All real variables are explicitly of Fortran type "double precision" (64 bit), giving a precision of 15 significant figures. Routines marked here with a `*` are those which have been explicitly parallelised using OpenMP.

9.3.1 Input files

Variables governing the model run are contained in two separate files, both of which are picked up by the main program file `q-gcm.F` via `INCLUDE` statements. These are the files to edit to change the model configuration from the default case.

parameter.src This contains dimensioning information for the model arrays, and defines the location of the oceanic domain within the atmospheric, and the relative resolutions of the two grids. It also contains the rotation parameters which define the latitude of the domains. The file also contains an explanation of all the parameters defined therein.

in_param.f A file defining the values of all other model parameters, both physical parameters, and control parameters for length of run, diagnostic intervals, etc. Also controls the directory to which output files are written. All definitions are in the form of Fortran assign statements. The file includes brief comments explaining the meaning of each parameter, including its units.

Both of these files are compiled as part of the program, so changes will only take effect after re-compilation. All these parameters (and many derived quantities) are written to standard output by the main program at the start of a run.

9.3.2 Program structure

A general idea of the structure of the code can be seen in the flow chart in figure 3. This shows the flow of control through the most important subroutines, and which equations they solve. Note that each `*.f` or `*.F` file may contain more than one subroutine (see the earlier list), but they are grouped logically, and well commented to describe their function.

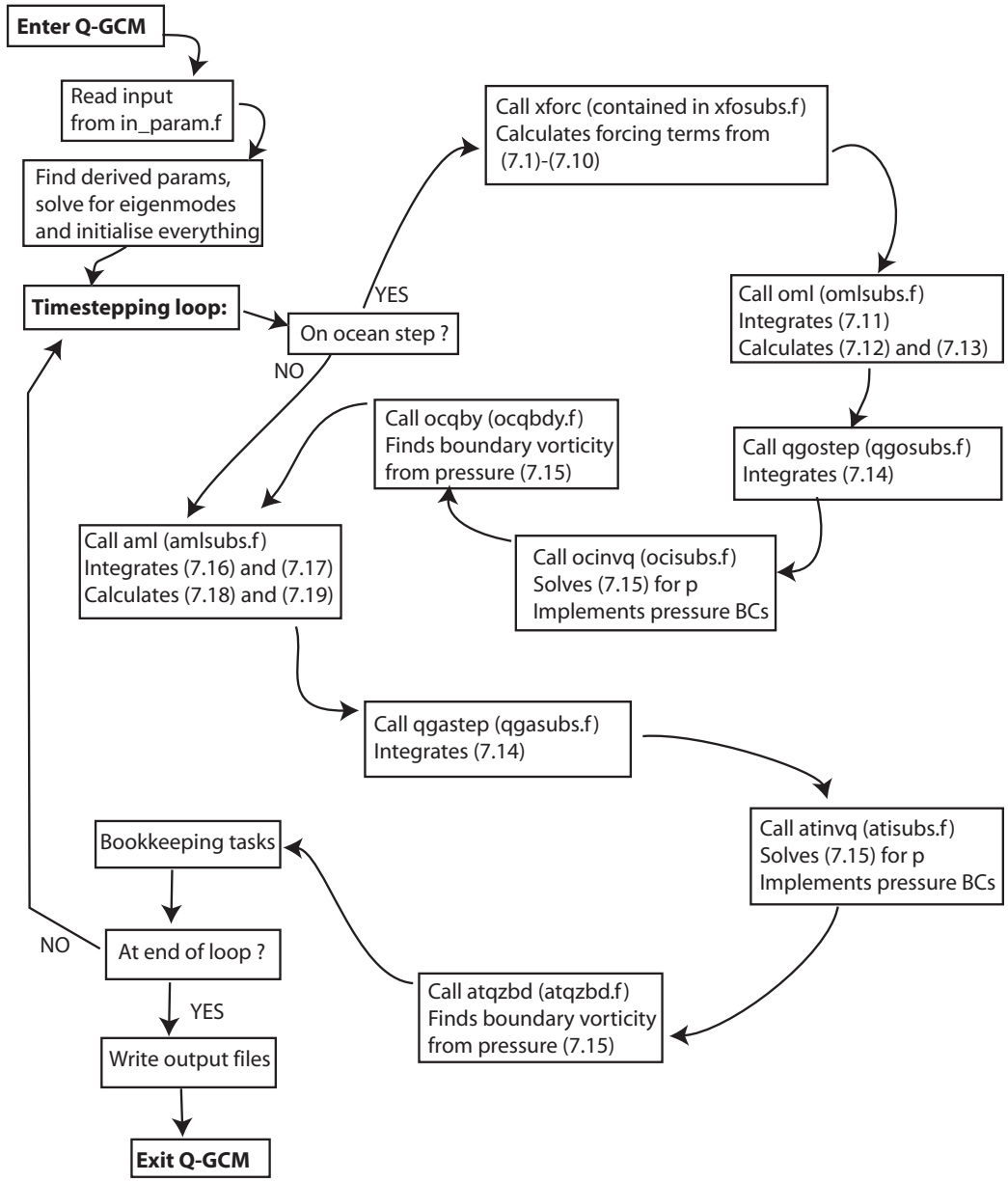

Figure 3: Flow chart showing the major subroutines of the code.

9.3.3 Output files

Output files are stored in the directory specified in the variable `outdir` on the first line of `in_param.f`. The filenames are hardwired into the code, and therefore can only be changed by altering the body of the code itself. Many of the file formats are in netCDF (`*.nc`) format, so that a description of variable names and units is included in the data file. The files are listed as follows:

input_parameters.m A file which includes all the parameters as a Matlab script. Simply run this script in Matlab, and variable names and values will be initialised.

restart.nc A netCDF file written periodically which includes the model state variables. This can be useful for restarting the run if the model crashes part-way through.

lastday.nc A netCDF file written at the end of the model run which includes the model state

variables and can be used for continuing the run.

monit.nc A netCDF format file which includes timeseries of some important quantities.

atast.nc, atpa.nc Two netCDF data files which describe the atmospheric state. The variables written to these files is controlled by the `outflag` array in `in_param.f`.

ocsst.nc, ocpo.nc Two netCDF data files which describe the oceanic state. The variables written to these files is controlled by the `outflag` array in `in_param.f`.

avges.nc A netCDF data file which includes data averaged over the entire run.

areas.nc A netCDF data file which includes timeseries data from selected regions.

covar.nc A netCDF data file which includes the data needed to calculate the covariance matrix for particular fields.

topog.nc A netCDF data file which includes the topography used for the run.

9.3.4 Location of variables

Where possible variables are held in common blocks, rather than being passed as arguments to subroutines. Each common block is defined in a separate `*.cmn` file, which contains comments describing the meaning of each variable, and its units. Each of the 16 common blocks contains a set of related variables thus:

`atconst.cmn`	Time independent atmospheric parameters
`athomog.cmn`	Homogeneous atmospheric solutions and their coefficients
`atmfft.cmn`	Atmospheric Fourier transform coefficients
`atnc.cmn`	Identifiers for netCDF dumps of atmospheric variables
`atstate.cmn`	Time varying atmospheric state (pressure, vorticity, entrainment and Ekman velocity)
`covar.cmn`	All variables needed for covsubs routines
`intrfac.cmn`	All mixed layer parameters and variables, for both atmosphere and ocean, including dynamic stresses
`monitor.cmn`	Diagnostic quantities for monitoring model
`monnc.cmn`	Identifiers for netCDF dumps of diagnostic variables
`occonst.cmn`	Time independent oceanic parameters
`ochomog.cmn`	Homogeneous oceanic solution and its coefficients
`ocnc.cmn`	Identifiers for netCDF dumps of oceanic variables
`ocnfft.cmn`	Oceanic Fourier sine transform coefficients
`ocstate.cmn`	Time varying oceanic state (pressure, vorticity, entrainment and Ekman velocity)
`radiate.cmn`	Input radiative parameters, and the derived coefficients needed for the radiation scheme
`timavge.cmn`	Used for averaging fields over the course of a run

These files are included at the start of those subroutines where they are required. They are all included in the main program, to ensure that the values of their contents remain defined at all times. Variables declared within subroutines are purely local, so "stack" storage can be used to save memory.

9.3.5 Diagnostics

Where possible, instantaneous diagnostic quantities are computed by subroutine `diagno`, whose calling frequency is determined by the variable `dgnday`, set in `in_param.f`. One exception to

this is that some quantities which would require inconvenient duplication of code are computed by the timestepping routines. Another exception is that convection-related quantities need to be computed in the timestepping subroutines, before convective adjustment is finally applied. These diagnostic quantities are all stored in the common block `monitor.cmn`, and output in netCDF format by subroutine `monnc_out`. Time averaged quantities are accumulated by subroutines `tavatm` and `tavocn` in the common block `timavge.cmn`, and finally computed and output by subroutine `tavout`.

9.3.6 Test programs

The code distribution includes three test programs. They were originally written to independently test particular routines, but can also be used to test the effects of changing certain input parameters, without needing to run the full Q-GCM model.

eigtest.f Tests subroutine `eigmod`, and thus the dependence of phase speeds, Rossby radii, damping timescales etc. on layer thicknesses, reduced gravities, diffusivities etc., as specified via `in_param.f`.

radtest.f Tests `radsubs.f`, and thus the dependence of the mean state radiative balance and the linearised radiation coefficients on the optical properties and radiative forcing, as specified via `in_param.f`.

toptest.F Tests `topsubs.F`, and thus enables the user to experiment with defining suitable topographies for the ocean and atmosphere. Creates a `topog.nc` file in the current directory.

These programs are compiled using the `Makefile` in the usual way, except that they don't use the optimisation and possible parallelisation options used for the full model.

9.3.7 Default case

The input files are supplied set to run a standard spin-up case with an ocean of 385×481 points and resolution 10 km, and an atmosphere of 128×65 points and resolution 120 km. This default case has 3 QG layers in both atmosphere and ocean, and requires 110 MB of static memory, or 340 MB if the covariance is computed.

9.3.8 Possible problems

- The program uses significant "stack" memory which may exceed the default limit on your system causing a `Segmentation fault` error. You will need to reset the limit by doing e.g.

  `limit stacksize 96m`, or even

  `limit stacksize unlimited` (C shell)

  `ulimit -s 98304` (bash or ksh; units are kbytes)

- When running the program in parallel, you may need to increase the limit on stack allocation by each separate thread. The name of the environment variable depends on the compiler being used; the syntax for setting it depends on the shell in the usual way. Examples are:

  `setenv MP_STACK_SIZE 40000000` (Csh, Compaq)

  `setenv XLSMPOPTS "stack40000000"=` (Csh, IBM)

  `setenv MP_SLAVE_STACKSIZE 40000000` (Csh, SGI)

  `setenv STACKSIZE 40000000` (Csh, Sun)

`setenv MPSTKZ 40000000` (linux/Csh, Portland Group Compiler)

`setenv KMP_STACKSIZE 40000000` (linux/Csh, Intel compiler)

Remember to set the number of threads required; syntax for this depends only on the shell:

`setenv OMP_NUM_THREADS 4` (Csh)

`export OMP_NUM_THREADS=4` (bash)

- Some systems (linux only?) seem to have an additional limit (typically a few Mbytes, and perhaps set by the kernel) on the allocation of stack storage, which we don't yet know how to reset. As a workaround, we have provided the `small_local` preprocessor option which causes the main temporary arrays in subroutines to be allocated as static rather than stack storage, by using the Fortran `SAVE` statement. If the enviroment settings described above do not fix any segmentation problems you encounter, try using `small_local` as described in section 9.2.

9.3.9 Feedback

The model conforms strictly to Fortran standards, and so should be highly portable. It has been checked using the public domain Fortran77 analyzer "ftnchek", which we highly recommend. The home website is:
`http://dsm.dsm.fordham.edu/~ftnchek/`.
This software is also available from the Netlib software repository and its many mirror sites.

Please report any compilation or runtime problems, with details of the operating system and compiler used, and whether running in parallel or not.

9.4 Changes Log

The following changes have been made from version 1.1:

- Addition of ∇_H^2 viscosity in the ocean.
- Addition of ∇_H^4 diffusion in SST and AST.
- Nondimensionalising the mixed boundary condition coefficient.
- Generalisation of atmosphere from 2 layers to N layers.
- Simplification of atmospheric radiation scheme.
- Addition of a minimum mixed layer thickness to the atmospheric mixed layer scheme.
- Added vorticity and Ekman pumping to time-average diagnostics. Added mean outgoing longwave radiation, mean pressure and potential vorticity to monitoring diagnostics.
- Addition of topography to both atmosphere and ocean components of the model.
- More explicit parallelisation for improved performance.
- More use of `#ifdef`s to reduce wasted data output in uncoupled simulations.
- Alterations to the structure which move more subroutines into separate `*.f/*.F` files, and shorten the main `q-gcm.F` file.
- Addition of `#ifdef small_local` flag to fix problems which arise on machines with a small predefined stacklimit.

Acknowledgements

Neil Stringfellow of Manchester Computing assisted with optimisation and parallelisation of the code. Peter Challenor assisted us to implement the statistics packages which are built into the code.

A Writing radiation as a perturbation to mean state

A.1 Functional form of radiation parameters

We follow a simplified, linearised model of vertical radiative heat transfer in the atmosphere (see Peixoto and Oort, 1992, for details on the basics of radiative balance). At any point in the atmosphere transfer is due to two sources: (i) blackbody radiative emission from radiatively active elements within the current layer, which is of the form

$$\text{Flux} = \int B(z)\tau_z(z, Z)\, dz$$

and (ii) radiation from other layers which is partially absorbed within the current layer

$$\text{Flux} = F_{\text{in}}\tau(Z_{k-1}, Z_k).$$

Here we have used a new notation Z_k for the unperturbed interface height coordinate (as the model is framed in terms of layer thicknesses) and introduced the transmissivity τ which is written as

$$\tau(z, Z) = e^{-\frac{Z-z}{z_0}}, \tag{A.1}$$

where $\tau(z, Z)$ is the fraction of radiation from z reaching Z, and z_0 is the (constant) optical depth of the layer (or $\tau(Z, z)$ is the fraction reaching Z from z for downgoing radiation). Derivatives of the transmissivities are then

$$\tau_z(z, Z) = \frac{\tau(z, Z)}{z_0}. \tag{A.2}$$

In a comprehensive model, the optical depth z_0 will vary with height. We assume for our model that we know the optical depth in each QG layer (z_k). Note also that we use an optical depth of zero at the bottom of the mixed layer, and z_m near the top of the mixed layer.

We have also used the blackbody radiation emitted from each layer which is written according to the Stefan-Boltzmann law

$$B_m(z) = \frac{\sigma}{2}(\overline{{}^aT_m} + {}^aT_m{}' - \gamma z)^4, \tag{A.3}$$

for the mixed layer, and

$$B_k(z) = \frac{\sigma}{2}({}^aT_k - \gamma z)^4, \tag{A.4}$$

for other layers. The terms in the temperature function represent constant potential temperature, the variation of in situ temperature with height which depends upon the adiabatic lapse rate γ.

A.2 Rules for linearising

We want to linearise the radiation equations about a mean state. To do this we expand about a mean and retain the first order terms in ${}^aT_m{}'$, ${}^a\eta_m$ and ${}^a\eta_k$. Note that the topography ${}^aD$ is included as a first order perturbation to the radiation, as it is to the dynamics. It therefore does not enter the mean state calculation. For the mixed layer radiation function

$$\frac{\sigma}{2}(\overline{{}^aT_m} - \gamma z + {}^aT_m{}')^4 \approx \frac{\sigma}{2}(\overline{{}^aT_m} - \gamma z)^4 + 2\sigma(\overline{{}^aT_m} - \gamma z)^3\, {}^aT_m{}'. \tag{A.5}$$

Transmissivities are expanded by writing, for example,

$$\begin{aligned}\tau({}^a h_m + {}^a D, {}^a h_1) &= e^{-\frac{{}^a Z_1 + {}^a\eta_1 - {}^a H_m - {}^a D - {}^a\eta_m}{z_1}} \\ &= e^{-\frac{{}^a Z_1 - {}^a H_m}{z_1}} e^{-\frac{{}^a\eta_1}{z_1}} e^{\frac{{}^a\eta_m + {}^a D}{z_1}} \\ &\approx \tau_1 \left(1 - \frac{{}^a\eta_1}{z_1}\right)\left(1 + \frac{{}^a\eta_m + {}^a D}{z_1}\right) \\ &\approx \tau_1 \left(1 - \frac{{}^a\eta_1}{z_1} + \frac{{}^a\eta_m + {}^a D}{z_1}\right)\end{aligned} \tag{A.6}$$

where $\tau_1 \equiv e^{-\frac{{}^a Z_1 - {}^a H_m}{z_1}}$ is the unperturbed value. For other layers, $\tau_k \equiv e^{-\frac{{}^a H_k}{z_k}}$.

Any integrals are also linearised as, for example,

$$\int_{{}^a h_m + {}^a D}^{{}^a Z_1 + {}^a\eta_1} f(z)dz \approx \int_{{}^a H_m}^{{}^a Z_1} f(z)dz - f({}^a H_m)({}^a\eta_m + {}^a D) + f({}^a Z_1){}^a\eta_1. \tag{A.7}$$

When calculating the radiation at any point, we write the full radiation balance, first expand the integration and then expand remaining terms, eliminating quadratic and higher order terms. This is explicitly done for each layer in the following section.

A.3 Calculations

Oceanic Mixed Layer absorbs all downgoing longwave radiation (as well as incoming shortwave radiation) which passes through the atmospheric mixed layer. Emits radiation according to its blackbody temperature,

$$F_0^{\uparrow} = B_0(0) = \sigma\left(\overline{{}^o T_m} + {}^o T_m{}'\right)^4. \tag{A.8}$$

We linearise this by writing

$$F_0^{\uparrow} \approx \overline{F_0^{\uparrow}} + D_0^{\uparrow}{}^o T_m{}' \tag{A.9}$$

where

$$\overline{F_0^{\uparrow}} = \sigma \overline{{}^o T_m}^4, \tag{A.10}$$

$$D_0^{\uparrow} = 4\sigma \overline{{}^o T_m}^3. \tag{A.11}$$

Atmospheric Mixed Layer To calculate upward radiation leaving the atmospheric mixed layer, we note that optical depth at the base of the mixed layer is assumed to be zero: thus outgoing radiation here must be internally generated:

$$F_m^{\uparrow} = \int_{{}^a D}^{{}^a h_m + {}^a D} B_m(z)\tau_z(z, {}^a h_m + {}^a D)\, dz. \tag{A.12}$$

We first linearise the integral, giving

$$\begin{aligned}F_m^{\uparrow} \approx \int_0^{{}^a H_m} B_m(z)\tau_z(z, {}^a h_m)\, dz + B_m({}^a H_m)\tau_z({}^a H_m, {}^a h_m + {}^a D)({}^a\eta_m + {}^a D) - \\ B_m(0)\tau_z(0, {}^a h_m + {}^a D){}^a D,\end{aligned} \tag{A.13}$$

and then evaluate each of the terms (noting that the last term disappears because the mixed layer is opaque at the bottom),

$$\begin{aligned}F_m^{\uparrow} \approx \int_0^{{}^a H_m} \left(\frac{\sigma}{2}(\overline{{}^a T_m} - \gamma z)^4 + 2\sigma\left(\overline{{}^a T_m} - \gamma z\right)^3 {}^a T_m{}'\right) \frac{e^{-\frac{{}^a H_m - z}{z_m}}}{z_m}\left(1 - \frac{{}^a\eta_m + {}^a D}{z_m}\right) dz \\ + \frac{\sigma}{2}\left(\overline{{}^a T_m} - \gamma {}^a H_m\right)^4 \frac{{}^a\eta_m + {}^a D}{z_m}.\end{aligned} \tag{A.14}$$

By omitting the remaining quadratic terms we obtain an expression for upwards radiation,

$$F_m^{\uparrow} \approx \frac{\sigma I_m^{\uparrow}}{2z_m} + \left(\frac{2\sigma}{z_m} \int_0^{{}^aH_m} \left(\overline{{}^aT_m} - \gamma z \right)^3 e^{-\frac{{}^aH_m - z}{z_m}} \, dz \right) {}^aT_m{}' + \left(\frac{\sigma}{2z_m} (\overline{{}^aT_m} - \gamma {}^aH_m)^4 - \frac{\sigma I_m^{\uparrow}}{2{z_m}^2} \right) ({}^a\eta_m + {}^aD). \quad \text{(A.15)}$$

Here we have defined

$$I_m^{\uparrow} = \int_0^{{}^aH_m} \left(\overline{{}^aT_m} - \gamma z \right)^4 e^{-\frac{{}^aH_m - z}{z_m}} \, dz,$$

and we use similar integrals when calculating radiation in other layers. The complete radiation formula is written in terms of the coefficients

$$\overline{F_m^{\uparrow}} = \frac{\sigma I_m^{\uparrow}}{2z_m}, \quad \text{(A.16)}$$

$$B_m^{\uparrow} = \left(\frac{\sigma}{2} (\overline{{}^aT_m} - \gamma {}^aH_m)^4 - \overline{F_m^{\uparrow}} \right) \frac{1}{z_m}, \quad \text{(A.17)}$$

$$C_m^{\uparrow} = B_m^{\uparrow}, \quad \text{(A.18)}$$

$$D_m^{\uparrow} = \frac{2\sigma}{z_m} \int_0^{{}^aH_m} \left(\overline{{}^aT_m} - \gamma z \right)^3 e^{-\frac{{}^aH_m - z}{z_m}} \, dz, \quad \text{(A.19)}$$

giving

$$F_m^{\uparrow} \approx \overline{F_m^{\uparrow}} + B_m^{\uparrow} {}^a\eta_m + C_m^{\uparrow} {}^aD + D_m^{\uparrow} {}^aT_m{}'. \quad \text{(A.20)}$$

These constant coefficients are evaluated at the initialisation of the model, and stored for later use.

Downgoing radiation can be calculated simply using the assumption that optical depth at the bottom of the mixed layer is $z_0 = 0$, so that radiation only depends upon local temperature,

$$\begin{aligned} F_m^{\downarrow} &\approx -\frac{\sigma}{2} \overline{{}^aT_m}^4 - 2\sigma \overline{{}^aT_m}^3 {}^aT_m{}' \\ &= \overline{F_m^{\downarrow}} + D_m^{\downarrow} {}^aT_m{}', \end{aligned} \quad \text{(A.21)}$$

where

$$\overline{F_m^{\downarrow}} = -\frac{\sigma}{2} \overline{{}^aT_m}^4, \quad \text{(A.22)}$$

$$D_m^{\downarrow} = -2\sigma \overline{{}^aT_m}^3. \quad \text{(A.23)}$$

Atmospheric QG layers The QG layers include both partial absorption and emmission terms,

$$F_k^{\uparrow} = F_{k-1}^{\uparrow} \tau({}^aZ_{k-1} + {}^a\eta_{k-1}, {}^aZ_k + {}^a\eta_k) + \int_{{}^aZ_{k-1} + {}^a\eta_{k-1}}^{{}^aZ_k + {}^a\eta_k} B_k(z) \tau_z(z, {}^aZ_k + {}^a\eta_k) \, dz. \quad \text{(A.24)}$$

Note that when $k = 1$, we use ${}^aZ_{k-1} = {}^aH_m$ and ${}^a\eta_{k-1} = {}^a\eta_m + {}^aD$. The full equation is thus

$$\begin{aligned} F_k^{\uparrow} = \left(\overline{F_{k-1}^{\uparrow}} + F_{k-1}^{\uparrow}{}' \right) \tau_k \left(1 + \frac{{}^a\eta_{k-1}}{z_k} - \frac{{}^a\eta_k}{z_k} \right) \\ + \int_{{}^aZ_{k-1}}^{{}^aZ_k} \frac{\sigma}{2} ({}^aT_k - \gamma z)^4 \frac{e^{-\frac{{}^aZ_k - z}{z_k}}}{z_k} \left(1 - \frac{{}^a\eta_k}{z_k} \right) dz \\ - \frac{\sigma}{2} ({}^aT_k - \gamma {}^aZ_{k-1})^4 \frac{\tau_k}{z_k} {}^a\eta_{k-1} + \frac{\sigma}{2} ({}^aT_k - \gamma {}^aZ_k)^4 \frac{{}^a\eta_k}{z_k}. \end{aligned} \quad \text{(A.25)}$$

Again we organise this equation into mean and perturbation quantities,

$$F_k^{\uparrow} \approx \overline{F_{k-1}^{\uparrow}}\tau_k + \frac{\sigma I_k^{\uparrow}}{2z_k} + {F_{k-1}^{\uparrow}}'\tau_k + \left(\overline{F_{k-1}^{\uparrow}} - \frac{\sigma}{2}({}^aT_k - \gamma^a Z_{k-1})^4\right)\frac{\tau_k}{z_k}{}^a\eta_{k-1} + \left(-\overline{F_{k-1}^{\uparrow}}\tau_k - \frac{\sigma I_k^{\uparrow}}{2z_k} + \frac{\sigma}{2}({}^aT_k - \gamma^a Z_k)^4\right)\frac{{}^a\eta_k}{z_k}, \quad \text{(A.26)}$$

where

$$I_k^{\uparrow} = \int_{{}^aZ_{k-1}}^{{}^aZ_k} \left(\overline{{}^aT_k} - \gamma z\right)^4 e^{-\frac{{}^aZ_k - z}{z_k}}\, dz.$$

Note that this equation requires contributions from layers below so that the individual terms need to be evaluated from the bottom upwards. The general form of the equation is

$$F_k^{\uparrow} \approx \overline{F_k^{\uparrow}} + \sum_{i=1}^{\min(k,N-1)} A_{k,i}^{\uparrow}{}^a\eta_i + B_k^{\uparrow a}\eta_m + C_k^{\uparrow a}D + D_k^{\uparrow a}{T_m}' \qquad \text{for } k = 1, N, \quad \text{(A.27)}$$

where

$$\overline{F_k^{\uparrow}} = \overline{F_{k-1}^{\uparrow}}\tau_k + \frac{\sigma I_k^{\uparrow}}{2z_k}, \quad \text{(A.28)}$$

$$D_1^{\uparrow} = D_m^{\uparrow}\tau_1, \quad \text{(A.29)}$$

$$D_k^{\uparrow} = D_{k-1}^{\uparrow}\tau_k \qquad \text{for } k = 2, N. \quad \text{(A.30)}$$

Calculation of the other two terms is more complicated:

$$B_1^{\uparrow} = \left(B_m^{\uparrow} + \frac{\overline{F_m^{\uparrow}}}{z_1} - \frac{\sigma}{2z_1}({}^aT_1 - \gamma^a H_m)^4\right)\tau_1, \quad \text{(A.31)}$$

$$B_k^{\uparrow} = B_{k-1}^{\uparrow}\tau_k \qquad \text{for } k = 2, N. \quad \text{(A.32)}$$

$$C_1^{\uparrow} = \left(C_m^{\uparrow} + \frac{\overline{F_m^{\uparrow}}}{z_1} - \frac{\sigma}{2z_1}({}^aT_1 - \gamma^a H_m)^4\right)\tau_1, \quad \text{(A.33)}$$

$$C_k^{\uparrow} = C_{k-1}^{\uparrow}\tau_k \qquad \text{for } k = 2, N. \quad \text{(A.34)}$$

$$A_{1,1}^{\uparrow} = \left(-\overline{F_m^{\uparrow}}\tau_1 - \frac{\sigma I_1^{\uparrow}}{2z_1} + \frac{\sigma}{2}({}^aT_1 - \gamma^a Z_1)^4\right)\frac{1}{z_1}, \quad \text{(A.35)}$$

$$A_{k,k}^{\uparrow} = \left(-\overline{F_{k-1}^{\uparrow}}\tau_k - \frac{\sigma I_k^{\uparrow}}{2z_k} + \frac{\sigma}{2}({}^aT_k - \gamma^a Z_k)^4\right)\frac{1}{z_k} \qquad \text{for } k = 2, N-1, \quad \text{(A.36)}$$

$$A_{k,k-1}^{\uparrow} = \left(\frac{\overline{F_{k-1}^{\uparrow}}}{z_k} + A_{k-1,k-1}^{\uparrow} - \frac{\sigma}{2z_k}({}^aT_k - \gamma^a Z_{k-1})^4\right)\tau_k \qquad \text{for } k = 2, N. \quad \text{(A.37)}$$

$$A_{k,i}^{\uparrow} = A_{k-1,i}^{\uparrow}\tau_k \qquad \text{for } i = 1, k-2 \quad \text{and} \quad k = 3, N. \quad \text{(A.38)}$$

For downgoing radiation the full equation is

$$F_k^{\downarrow} = F_{k+1}^{\downarrow}\tau({}^aZ_{k-1} + {}^a\eta_{k-1}, {}^aZ_k + {}^a\eta_k) + \int_{{}^aZ_{k-1}+{}^a\eta_{k-1}}^{{}^aZ_k+{}^a\eta_k} B_k(z)\tau_z({}^aZ_{k-1} + {}^a\eta_{k-1}, z)\, dz \quad \text{(A.39)}$$

which reduces to

$$F_k^{\downarrow} \approx \left(\overline{F_{k+1}^{\downarrow}} + {F_{k+1}^{\downarrow}}'\right)\tau_k \left(1 + \frac{{}^a\eta_{k-1}}{z_k} - \frac{{}^a\eta_k}{z_k}\right) - \int_{{}^aZ_{k-1}}^{{}^aZ_k} \frac{\sigma}{2}({}^aT_k - \gamma z)^4 \frac{e^{-\frac{z-{}^aZ_{k-1}}{z_k}}}{z_k}\left(1 + \frac{{}^a\eta_{k-1}}{z_k}\right) dz + \frac{\sigma}{2}({}^aT_k - \gamma^a Z_{k-1})^4 \frac{{}^a\eta_{k-1}}{z_k} - \frac{\sigma}{2}({}^aT_k - \gamma^a Z_k)^4 \frac{\tau_k}{z_k}{}^a\eta_k, \quad \text{(A.40)}$$

giving

$$F_k^{\downarrow} \approx \overline{F_{k+1}^{\downarrow}}\tau_k - \frac{\sigma I_k^{\downarrow}}{2z_k} + {F_{k+1}^{\downarrow}}'\tau_k + \left(\overline{F_{k+1}^{\downarrow}}\tau_k - \frac{\sigma I_k^{\downarrow}}{2z_k} + \frac{\sigma}{2}({}^aT_k - \gamma^a Z_{k-1})^4\right)\frac{{}^a\eta_{k-1}}{z_k} + \left(-\overline{F_{k+1}^{\downarrow}} - \frac{\sigma}{2}({}^aT_k - \gamma^a Z_k)^4\right)\frac{\tau_k}{z_k}{}^a\eta_k, \quad \text{(A.41)}$$

where

$$I_k^{\downarrow} = \int_{{}^aZ_{k-1}}^{{}^aZ_k} \left(\overline{{}^aT_k} - \gamma z\right)^4 e^{-\frac{z-{}^aZ_{k-1}}{z_k}}\, dz.$$

Again this equation requires contributions from layers above, so that the individual terms need to be evaluated from the top downwards. This can be written in the general form

$$F_k^{\downarrow} \approx \overline{F_k^{\downarrow}} + \sum_{i=\max(k-1,1)}^{N-1} A_{k,i}^{\downarrow}{}^a\eta_i + B_k^{\downarrow a}\eta_m + C_k^{\downarrow a}D \qquad \text{for } k = 1, N, \quad \text{(A.42)}$$

where we know

$$\overline{F_k^{\downarrow}} = \overline{F_{k+1}^{\downarrow}}\tau_k - \frac{\sigma I_k^{\downarrow}}{2z_k}, \quad \text{(A.43)}$$

$$B_1^{\downarrow} = \left(\overline{F_2^{\downarrow}}\tau_1 - \frac{\sigma I_1^{\downarrow}}{2z_1} + \frac{\sigma}{2}({}^aT_1 - \gamma^a H_m)^4\right)\frac{1}{z_1}, \quad \text{(A.44)}$$

$$B_k^{\downarrow} = 0 \qquad \text{for } k = 2, N, \quad \text{(A.45)}$$

$$C_1^{\downarrow} = B_1^{\downarrow} \quad \text{(A.46)}$$

$$C_k^{\downarrow} = 0 \qquad \text{for } k = 2, N, \quad \text{(A.47)}$$

$$A_{N,N-1}^{\downarrow} = \left(-\frac{\sigma I_N^{\downarrow}}{2z_N} + \frac{\sigma}{2}({}^aT_N - \gamma^a Z_{N-1})^4\right)\frac{1}{z_N}, \quad \text{(A.48)}$$

$$A_{k,k-1}^{\downarrow} = \left(\overline{F_{k+1}^{\downarrow}}\tau_k - \frac{\sigma I_k^{\downarrow}}{2z_k} + \frac{\sigma}{2}({}^aT_k - \gamma^a Z_{k-1})^4\right)\frac{1}{z_k} \qquad \text{for } k = 2, N-1, \quad \text{(A.49)}$$

$$A_{k,k}^{\downarrow} = \left(A_{k+1,k}^{\downarrow} - \frac{\overline{F_{k+1}^{\downarrow}}}{z_k} - \frac{\sigma}{2z_k}({}^aT_k - \gamma^a Z_k)^4\right)\tau_k, \qquad \text{for } k = 1, N-1, \quad \text{(A.50)}$$

$$A_{k,i}^{\downarrow} = A_{k+1,i}^{\downarrow}\tau_k \qquad \text{for } k = 1, N-1 \text{ and } i = k+1, N-1, \quad \text{(A.51)}$$

B Constraints on mass and momentum

B.1 Ocean constraints

The area integrated changes in layer heights in the ocean need to be constrained, as they are not calculated explicitly. In layers 2 to N, where ${}^{o}e_k = 0$ we simply apply the condition

$$\iint {}^{o}\eta_{kt}\, dA = 0, \qquad k > 1. \tag{B.1}$$

At interface 1, however, entrainment is finite. To conserve mass one would need the condition

$$\iint {}^{o}\eta_{1t}\, dA = -\iint {}^{o}e_1\, dA. \tag{B.2}$$

This equation, as well as representing the conservation of mass, also determines the amount of heat entering the deep ocean. Without any overturning circulation or vertical diffusion there is no mechanism for heat to be transferred from the deep ocean back to the surface layer, and so the above condition implies a drift in deep ocean temperature. Therefore, we instead apply the constraint

$$\iint {}^{o}\eta_{1t}\, dA = 0. \tag{B.3}$$

There is no momentum constraint to satisfy as the basin is closed.

B.2 Atmospheric momentum constraints

Following McWilliams (1977) we need to find constraints on momentum for flow in the channel atmosphere. We start with an example of using constraints in the continuous case and then progress to the discrete case.

B.2.1 The continuous case

To determine these constraints in the continuous case, we integrate as an example the atmospheric layer 1 vorticity from (7.14) over the entire basin area A,

$$\iint \nabla_H^2 {}^{a}p_{1t}\, dA - \frac{f_0^2}{{}^{a}H_1} \iint {}^{a}\eta_{1t}\, dA = \iint J({}^{a}q_1, {}^{a}p_1)\, dA + \frac{f_0^2}{{}^{a}H_1} \iint ({}^{a}e_1 - {}^{a}w_{ek})\, dA - {}^{a}A_4 \iint \nabla_H^6 {}^{a}p_1\, dA. \tag{B.4}$$

We apply a mass constraint

$$\iint {}^{a}\eta_{1t}\, dA = -\iint {}^{a}e_1\, dA, \tag{B.5}$$

and require that there is no net advection over the basin,

$$\iint \nabla_H^2 {}^{a}p_{1t}\, dA = -\frac{f_0^2}{{}^{a}H_1} \iint {}^{a}w_{ek}\, dA - {}^{a}A_4 \iint \nabla_H^6 {}^{a}p_1\, dA. \tag{B.6}$$

This equation can be simplified by writing

$$\iint \nabla_H.(\nabla_H {}^{a}p_{1t})\, dA = -\frac{f_0}{{}^{a}H_1} \iint \nabla_H \times \tau\, dA - {}^{a}A_4 \iint \nabla_H.\nabla_H(\nabla_H^4 {}^{a}p_1)\, dA \tag{B.7}$$

so that application of Gauss' and Stokes' theorems in a periodic channel gives

$$\int \left[{}^{a}p_{1yt}\right]_0^{L_y}\, dx = \frac{f_0}{{}^{a}H_1} \int \left[{}^{a}\tau^{x}\right]_0^{L_y}\, dx - {}^{a}A_4 \int \left[\nabla_H^4 {}^{a}p_{1y}\right]_0^{L_y}\, dx, \tag{B.8}$$

where the square brackets refers to the difference between the values of the variables within at the north and south of the domain. The equivalent equations can be found in other layers in the same way.

B.2.2 The discrete case

The same equations can be written for the discrete case, however we only know about vorticity evolution at internal points in the domain (and not the boundary points). Therefore we integrate only over the internal points (area A_I)

$$\iint \nabla_H^2 {}^a p_{1t}\, dA_I - \frac{f_0^2}{{}^a H_1} \iint {}^a \eta_{1t}\, dA_I = \iint J({}^a q_1, {}^a p_1)\, dA_I + \frac{f_0^2}{{}^a H_1} \iint ({}^a e_1 - {}^a w_{ek})\, dA_I - {}^a A_4 \iint \nabla_H^6 {}^a p_1\, dA_I \quad \text{(B.9)}$$

and note that there remain finite contributions from the advection terms and the mass balance. The displacement and entrainment terms are simplified using

$$\iint ({}^a \eta_{1t} + {}^a e_1)\; dA = \iint ({}^a \eta_{1t} + {}^a e_1)\; dA_I + \frac{\Delta y}{2} \int \left[({}^a \eta_{1t} + {}^a e_1)^{(1)} + ({}^a \eta_{1t} + {}^a e_1)^{(n)} \right] dx = 0, \quad \text{(B.10)}$$

giving

$$\iint ({}^a \eta_{1t} + {}^a e_1)\; dA_I = -\frac{\Delta y}{2} \int \left[({}^a \eta_{1t} + {}^a e_1)^{(1)} + ({}^a \eta_{1t} + {}^a e_1)^{(n)} \right] dx. \quad \text{(B.11)}$$

Here the bracketed superscripts refer to the value of the quantity at that gridpoint.

Since the Jacobian describes conservative advection of q_1, its integral vanishes over the interior of the domain, leaving contributions only from some near-boundary components. So we can write

$$\iint J({}^a q_1, {}^a p_1)\, dA_I = \int \left(J_1^{(1)} + J_1^{(n)} \right) dx. \quad \text{(B.12)}$$

where J_k is the evaluation of the non-cancelling Jacobian terms near the boundary of layer k.

Following McWilliams (1977) we choose to satisfy constraints along each of the north and south boundaries indpendently. We gather all the terms in (B.9), once again applying Stokes' and Gauss' theorem to simplify the pressure derivatives and Ekman pumping term, and write two equations for the constraints on the atmospheric layer 1 flow,

$$-\int {}^a p_{1\,yt}^{(1.5)}\, dx + \frac{f_0^2 \Delta y}{2\,{}^a H_1} \int {}^a \eta_{1\,t}^{(1)}\, dx = \int J_1^{(1)}\, dx - \frac{f_0^2 \Delta y}{2\,{}^a H_1} \int {}^a e_1{}^{(1)}\, dx - \frac{f_0}{{}^a H_1} \int {}^a \tau^{x(1.5)}\, dx + {}^a A_4 \int {}^a p_{1\,5y}^{(1.5)}\, dx, \quad \text{(B.13)}$$

$$\int {}^a p_{1\,yt}^{(n-0.5)}\, dx + \frac{f_0^2 \Delta y}{2\,{}^a H_1} \int {}^a \eta_{1\,t}^{(n)}\, dx = \int J_1^{(n)}\, dx - \frac{f_0^2 \Delta y}{2\,{}^a H_1} \int {}^a e_1{}^{(n)}\, dx + \frac{f_0}{{}^a H_1} \int {}^a \tau^{x(n-0.5)}\, dx - {}^a A_4 \int {}^a p_{1\,5y}^{(n-0.5)}\, dx. \quad \text{(B.14)}$$

Note that the integral over internal pressure points gives terms which must be evaluated half a gridpoint in from the boundary (the first and last two terms in the above equations).

These equations can be generalised to the matrix form, so that the constraints on all layers can be written

$$-\int \mathbf{p}_{yt}^{(1.5)}\, dx + \frac{f_0^2 \Delta y}{2} \mathsf{A} \int \mathbf{p}_t^{(1)}\, dx = \int \mathbf{J}^{(1)}\, dx + \frac{f_0^2 \Delta y}{2} {}^{\mathbf{a}}\mathsf{B} \int {}^{\mathbf{a}}\mathbf{e}^{(1)}\, dx + {}^a A_4 \int \mathbf{p}_{5y}^{(1.5)}\, dx, \quad \text{(B.15)}$$

$$\int \mathbf{p}_{yt}^{(n-0.5)}\,dx + \frac{f_0^2\Delta y}{2}\mathsf{A}\int \mathbf{p}_t^{(n)}\,dx = \int \mathbf{J}^{(n)}\,dx + \frac{f_0^2\Delta y}{2}{}^{\mathtt{a}}\mathsf{B}\int {}^{\mathtt{a}}\mathbf{e}^{(n)}\,dx - {}^{a}A_4\int \mathbf{p}_{5y}^{(n-0.5)}\,dx. \quad \text{(B.16)}$$

The right hand sides of (B.15) and (B.16) are accumulated in the program during timestepping. Therefore we define

$$\mathbf{L^S} = -\int \mathbf{p}_y^{(1.5)}\,dx + \frac{f_0^2\Delta y}{2}\mathsf{A}\int \mathbf{p}^{(1)}\,dx$$

$$\mathbf{L^N} = \int \mathbf{p}_y^{(n-0.5)}\,dx + \frac{f_0^2\Delta y}{2}\mathsf{A}\int \mathbf{p}^{(n)}\,dx$$

$$\mathbf{R^S} = \int \mathbf{J}^{(1)}\,dx + \frac{f_0^2\Delta y}{2}{}^{\mathtt{a}}\mathsf{B}\int {}^{\mathtt{a}}\mathbf{e}^{(1)}\,dx + {}^{a}A_4\int \mathbf{p}_{5y}^{(1.5)}\,dx,$$

$$\mathbf{R^N} = \int \mathbf{J}^{(n)}\,dx + \frac{f_0^2\Delta y}{2}{}^{\mathtt{a}}\mathsf{B}\int {}^{\mathtt{a}}\mathbf{e}^{(n)}\,dx - {}^{a}A_4\int \mathbf{p}_{5y}^{(n-0.5)}\,dx.$$

It follows from (B.15) and (B.16) that, after timestepping, the new values of $\mathbf{L}$ can be written

$$\mathbf{L}_{new} = \mathbf{L}_{old} + 2\delta t\mathbf{R}$$

for both the northern and southern boundaries.

With knowledge of the evolution of $\mathbf{L}$ at both the north and south boundaries we proceed to solve for pressure using the Helmholtz solver, which returns a modal pressure $\tilde{\mathbf{p}}$ which is zero on both north and south boundaries. We are looking for $\hat{\mathbf{p}}$, the full modal pressure including boundary contributions, which can be written as a linear superposition of $\tilde{\mathbf{p}}$ and two precomputed solutions to the homogeneous Helmholtz equation:

$$\hat{\mathbf{p}} = \tilde{\mathbf{p}} + \mathbf{C_1P_1} + \mathbf{C_2P_2}.$$

We need to solve for coefficient vectors $\mathbf{C_1}$ and $\mathbf{C_2}$.

To do this we take advantage of the conversion between modal and layer pressures to write

$$\mathbf{L^S} = -\int \mathsf{C}_{\mathrm{m2l}}\hat{\mathbf{p}}_y^{(1.5)}\,dx + \frac{f_0^2\Delta y}{2}\mathsf{A}\int \mathsf{C}_{\mathrm{m2l}}\hat{\mathbf{p}}^{(1)}\,dx,$$

which gives

$$\mathbf{L^S} = -\mathsf{C}_{\mathrm{m2l}}\int \left(\tilde{\mathbf{p}}_y^{(1.5)} + \mathbf{C_1}{\mathbf{P_1}}_y^{(1.5)} + \mathbf{C_2}{\mathbf{P_2}}_y^{(1.5)}\right)dx + \frac{f_0^2\Delta y}{2}\mathsf{A}\mathsf{C}_{\mathrm{m2l}}\int \left(\tilde{\mathbf{p}}^{(1)} + \mathbf{C_1}{\mathbf{P_1}}^{(1)} + \mathbf{C_2}{\mathbf{P_2}}^{(1)}\right)dx.$$

Premultiply by $\mathsf{C}_{\mathrm{l2m}}$,

$$\mathsf{C}_{\mathrm{l2m}}\mathbf{L^S} = -\int \left(\tilde{\mathbf{p}}_y^{(1.5)} + \mathbf{C_1}{\mathbf{P_1}}_y^{(1.5)} + \mathbf{C_2}{\mathbf{P_2}}_y^{(1.5)}\right)dx + \frac{f_0^2\Delta y}{2}\lambda\int \left(\tilde{\mathbf{p}}^{(1)} + \mathbf{C_1}{\mathbf{P_1}}^{(1)} + \mathbf{C_2}{\mathbf{P_2}}^{(1)}\right)dx,$$

where have used the identities $\mathsf{C}_{\mathrm{l2m}}\mathsf{C}_{\mathrm{m2l}} = 1$ and $\mathsf{C}_{\mathrm{l2m}}\mathsf{A}\mathsf{C}_{\mathrm{m2l}} = \lambda$. Thus, using the fact that $\tilde{\mathbf{p}}^{(\mathbf{1})}$ vanishes on the boundary, we can write

$$\mathsf{C}_{\mathrm{l2m}}\mathbf{L^S} + \int \tilde{\mathbf{p}}_y^{(1.5)}\,dx = \mathbf{C_1}\left[-\int {\mathbf{P_1}}_y^{(1.5)}\,dx + \frac{f_0^2\Delta y}{2}\lambda\int {\mathbf{P_1}}^{(1)}\,dx\right] + \mathbf{C_2}\left[-\int {\mathbf{P_2}}_y^{(1.5)}\,dx + \frac{f_0^2\Delta y}{2}\lambda\int {\mathbf{P_2}}^{(1)}\,dx\right]. \quad \text{(B.17)}$$

Likewise, for the northern boundary we can write

$$\mathsf{C}_{\mathrm{l2m}}\mathbf{L^N} - \int \tilde{\mathbf{p}}_y^{(n-0.5)}\,dx = \mathbf{C_1}\left[\int \mathbf{P_1}_y^{(n-0.5)}\,dx + \frac{f_0^2\Delta y}{2}\lambda\int \mathbf{P_1}^{(n)}\,dx\right] + \mathbf{C_2}\left[\int \mathbf{P_2}_y^{(n-0.5)}\,dx + \frac{f_0^2\Delta y}{2}\lambda\int \mathbf{P_2}^{(n)}\,dx\right]. \quad \text{(B.18)}$$

Thus we have a pair of simultaneous linear equations for the homogeneous coefficients $\mathbf{C_1}$ and $\mathbf{C_2}$. The terms in the square brackets can be precomputed at model initialisation, and the left hand sides are determined at each timestep, and $\mathbf{C_1}$ and $\mathbf{C_2}$ are then found. These are used to form the baroclinic modes, and the modes are transformed into layer pressures using the matrix $\mathsf{C}_{\mathrm{m2l}}$.

The barotropic mode is a special case of the above, because the eigenvalue λ_b is zero. In addition, according to McWilliams (1977) we need only one constraint on the barotropic mode; we choose to apply this on the southern boundary. In addition we only have one homogeneous barotropic mode (as the barotropic mode satisfies $p_{yy} = 0$). The equations then reduce to

$$\left(\mathsf{C}_{\mathrm{l2m}}\mathbf{L^S} + \int \tilde{\mathbf{p}}_y^{(1.5)}\,dx\right)_b = -C_b\int P_{b_y}^{(1.5)}\,dx,$$

where we have chosen $P_b = 1 - \frac{y}{{}^aY}$, and thus can explicitly write

$$C_b = \frac{{}^aY}{{}^aX}\left(\mathsf{C}_{\mathrm{l2m}}\mathbf{L^S} + \int \tilde{\mathbf{p}}_y^{(1.5)}\,dx\right)_b.$$

C Application of boundary conditions

The boundary conditions in the model are described by (2.26)–(2.28). To apply these conditions at (say) the southern boundary of the domain, we use the second order accurate centred-difference approximations to pressure derivatives on the boundary:

$$p_y^{(1)} = \frac{p^{(2)} - p^{(0)}}{2\Delta} \quad \text{(C.1)}$$

$$p_{yy}^{(1)} = \frac{p^{(2)} - 2p^{(1)} + p^{(0)}}{\Delta^2} \quad \text{(C.2)}$$

where $p^{(0)}$ is defined at a ghost point one gridlength outside the boundary, and Δ is the grid spacing. Using (2.27), and noting that the outward normal derivative $p_n = -p_y$,

$$\frac{p^{(2)} - 2p^{(1)} + p^{(0)}}{\Delta^2} = \frac{\alpha_{bc}}{\Delta}\frac{p^{(2)} - p^{(0)}}{2\Delta} \quad \text{(C.3)}$$

which can be solved for $p^{(0)}$:

$$p^{(0)} = \frac{(\frac{\alpha_{bc}}{2} - 1)p^{(2)} + 2p^{(1)}}{\frac{\alpha_{bc}}{2} + 1}. \quad \text{(C.4)}$$

Eliminating $p^{(0)}$, the boundary derivatives can thus be restated as

$$p_y^{(1)} = \frac{p^{(2)} - p^{(1)}}{\Delta(\frac{\alpha_{bc}}{2} + 1)} \quad \text{(C.5)}$$

$$p_{yy}^{(1)} = \frac{\alpha_{bc}(p^{(2)} - p^{(1)})}{\Delta^2(\frac{\alpha_{bc}}{2} + 1)}. \quad \text{(C.6)}$$

Likewise at the northern boundary we can write

$$p_y^{(n)} = \frac{p^{(n)} - p^{(n-1)}}{\Delta(\frac{\alpha_{bc}}{2} + 1)} \tag{C.7}$$

$$p_{yy}^{(n)} = \frac{\alpha_{bc}(p^{(n-1)} - p^{(n)})}{\Delta^2(\frac{\alpha_{bc}}{2} + 1)}. \tag{C.8}$$

At the western boundary in the ocean, the condition is identical to the southern condition (with y derivatives replaced by x), and eastern boundaries are equivalent to northern boundaries.

D Starting at radiative equilibrium

We can start at radiative equilibrium by writing

$${}^{a}F_m^T = 0 \tag{D.1}$$

$${}^{o}F_m^T = 0 \tag{D.2}$$

$${}^{a}e_1 = 0 \tag{D.3}$$

These can be used to find starting conditions for ${}^{a}T_m{}'$, ${}^{o}T_m{}'$ and ${}^{a}\eta_1$, and we assume ${}^{o}\eta_1 = 0$ and ${}^{a}\eta_m = 0$, and neglect atmospheric topography. We therefore set

$$0 = -\sum_{i=1}^{N-1} A_{1,i}^{\downarrow}\, {}^{a}\eta_i - D_m^{\uparrow}\, {}^{a}T_m{}' - F_s', \tag{D.4}$$

$$0 = (\lambda - D_m^{\downarrow}){}^{a}T_m{}' + (-\lambda - D_0^{\uparrow}){}^{o}T_m{}' - F_s', \tag{D.5}$$

$$0 = \sum_{i=1}^{N-1} \left(A_{1,i}^{\downarrow} - A_{N,i}^{\uparrow}\right) {}^{a}\eta_i + (D_m^{\uparrow} - D_2^{\uparrow}){}^{a}T_m{}'. \tag{D.6}$$

Here we have insufficient equations to constrain ${}^{a}\eta_i$ for all i, and so we further assume ${}^{a}\eta_i = 0$ for $i > 1$. Using (D.4) and (D.6) we get

$${}^{a}T_m{}' = \frac{(A_{1,1}^{\downarrow} - A_{N,1}^{\uparrow})F_s'}{A_{N,1}^{\uparrow}D_m^{\uparrow} - A_{1,1}^{\downarrow}D_2^{\uparrow}}, \tag{D.7}$$

and feed this back into (D.6):

$${}^{a}\eta_1 = \frac{(-D_m^{\uparrow} + D_2^{\uparrow})F_s'}{A_{N,1}^{\uparrow}D_m^{\uparrow} - A_{1,1}^{\downarrow}D_2^{\uparrow}} \tag{D.8}$$

We then find ocean mixed layer temperature from (D.5):

$${}^{o}T_m{}' = \left(\frac{(\lambda - D_m^{\downarrow})(A_{1,1}^{\downarrow} - A_{N,1}^{\uparrow})}{A_{N,1}^{\uparrow}D_m^{\uparrow} - A_{1,1}^{\downarrow}D_2^{\uparrow}} - 1\right) \frac{F_s'}{\lambda + D_0^{\uparrow}} \tag{D.9}$$

E Energy

The definitions of potential and kinetic energy in a QG model are specified by Holland (1978). We write the total kinetic energy in layer k as

$$\mathrm{KE}_k \equiv 0.5\rho H_k \iint ({u_k}^2 + {v_k}^2)\, dA,$$

while potential energy is defined in terms of interfaces rather than layers, and is simply

$$\mathrm{PE}_k \equiv 0.5\rho g'_k \iint (\eta_k)^2\, dA.$$

In addition we define the (positive definite) dissipations in layer k due to the Laplacian and biharmonic friction respectively as

$$\epsilon_{2k} \equiv -A_2\rho H_k \iint (u_k\nabla_H^2 u_k + v_k\nabla_H^2 v_k)\, dA.$$

$$\epsilon_{4k} \equiv A_4\rho H_k \iint (u_k\nabla_H^4 u_k + v_k\nabla_H^4 v_k)\, dA.$$

We start by considering the kinetic energy in atmosphere layer 1. Starting with (2.24) we multiply the atmospheric layer 1 component by ${}^a p_1$ and integrate over the entire domain,

$$\iint {}^a p_1 \left(\nabla_H^2 {}^a p_{1t} - \frac{f_0^2}{{}^a H_1} {}^a\eta_{1t}\right) dA = \frac{f_0^2}{{}^a H_1} \iint {}^a p_1({}^a e_1 - {}^a w_{ek})\, dA - {}^a A_4 \iint {}^a p_1 \nabla_H^6 {}^a p_1\, dA. \quad \text{(E.1)}$$

To write a kinetic energy equation we need to replace ${}^a p$ (which contains an arbitrary additive constant) with its derivatives (which are unambiguous). Therefore, we integrate by parts where possible,

$$\begin{aligned}
&\int \left([{}^a p_1 {}^a p_{1xt}]_0^{L_x} - \int {}^a p_{1_x} {}^a p_{1xt}\, dx\right) dy + \int \left([{}^a p_1 {}^a p_{1yt}]_0^{L_y} - \int {}^a p_{1_y} {}^a p_{1yt}\, dy\right) dx \\
&\quad - \frac{f_0^2}{{}^a H_1} \iint {}^a p_1 {}^a\eta_{1t}\, dA = \frac{f_0^2}{{}^a H_1} \iint {}^a p_1 {}^a e_1\, dA - \frac{f_0}{{}^a H_1} \int \left([{}^a p_1 {}^a\tau^y]_0^{L_x} - \int {}^a p_{1_x} {}^a\tau^y\, dx\right) dy \\
&+ \frac{f_0}{{}^a H_1} \int \left([{}^a p_1 {}^a\tau^x]_0^{L_y} - \int {}^a p_{1_y} {}^a\tau^x\, dy\right) dx - {}^a A_4 \int \left([{}^a p_1 \nabla_H^4 {}^a p_{1x}]_0^{L_x} - \int {}^a p_{1x} \nabla_H^4 {}^a p_{1x}\, dx\right) dy \\
&\quad - {}^a A_4 \int \left([{}^a p_1 \nabla_H^4 {}^a p_{1y}]_0^{L_y} - \int {}^a p_{1y} \nabla_H^4 {}^a p_{1y}\, dy\right) dx, \quad \text{(E.2)}
\end{aligned}$$

where we have expanded Ekman pumping terms into separate components of stress. We then write pressure gradients as geostrophic velocities and multiply by $\frac{{}^a\rho {}^a H_1}{f_0^2}$ to give

$$\begin{aligned}
&- {}^a\rho {}^a H_1 \iint ({}^a v_1 {}^a v_{1t} + {}^a u_1 {}^a u_{1t})\, dA - {}^a\rho \iint {}^a p_1 {}^a\eta_{1t}\, dA = {}^a\rho \iint {}^a p_1 {}^a e_1\, dA \\
&\quad + {}^a\rho \iint ({}^a v_1 {}^a\tau^y + {}^a u_1 {}^a\tau^x)\, dA + {}^a A_4 {}^a\rho {}^a H_1 \iint \left({}^a v_1 \nabla_H^4 {}^a v_1 + {}^a u_1 \nabla_H^4 {}^a u_1\right) dA \\
&\quad - \frac{{}^a\rho {}^a H_1}{f_0} \int \left[{}^a p_1 \left({}^a v_{1t} + \frac{{}^a\tau^y}{{}^a H_1} + {}^a A_4 \nabla_H^4 {}^a v_1\right)\right]_0^{L_x} dy \\
&\quad + \frac{{}^a\rho {}^a H_1}{f_0} \int \left[{}^a p_1 \left({}^a u_{1t} + \frac{{}^a\tau^x}{{}^a H_1} + {}^a A_4 \nabla_H^4 {}^a u_1\right)\right]_0^{L_y} dx. \quad \text{(E.3)}
\end{aligned}$$

The last two terms in this equation respectively represent jumps in quantities across the atmospheric domain in x and y. In the first case we note that jumps across a periodic domain must

be zero. The second integral can be cancelled using the momentum constraint (B.8). The result is the simplified equation

$$ {}^a\mathrm{KE}_{1t} + {}^a\rho \iint {}^a p_1 {}^a\eta_{1t}\, dA = -{}^a\rho \iint {}^a p_1 {}^a e_1\, dA - {}^a\rho \iint ({}^a v_1 {}^a\tau^y + {}^a u_1 {}^a\tau^x)\, dA - {}^a\epsilon_{41}. \quad \text{(E.4)} $$

Here the first term on the right represents the forcing due to up- or downwelling; the second term represents the drag on the bottom of the atmosphere and the third is the dissipation of kinetic energy by the biharmonic viscosity. The analogous equation for other layers in the atmosphere can be written

$$ {}^a\mathrm{KE}_{kt} + {}^a\rho \iint {}^a p_k\, ({}^a\eta_{kt} - {}^a\eta_{k-1t})\, dA = {}^a\rho \iint {}^a p_k ({}^a e_{k-1} - {}^a e_k)\, dA - {}^a\epsilon_{4k}. \quad \text{(E.5)} $$

It follows that by summing the equations for all layers we can write a total energy equation for the system,

$$ \begin{aligned} {}^a\mathrm{KE}_{1t} + {}^a\mathrm{KE}_{2t} + \ldots + {}^a\mathrm{KE}_{Nt} + {}^a\mathrm{PE}_{1t} + \ldots + {}^a\mathrm{PE}_{N-1t} &= -{}^a\rho \iint ({}^a v_1 {}^a\tau^y + {}^a u_1 {}^a\tau^x)\, dA \\ &- {}^a\rho \iint \left({}^a g'_1 {}^a\eta_1 {}^a e_1 + \ldots + {}^a g'_{N-1} {}^a\eta_{N-1} {}^a e_{N-1}\right) dA - {}^a\epsilon_{41} - {}^a\epsilon_{42} - \ldots - {}^a\epsilon_{4N}. \quad \text{(E.6)} \end{aligned} $$

where we have assumed that ${}^a e_N$ (entrainment at the top of the atmosphere) is always zero.

We repeat this exercise for the ocean, beginning with layer 1 where each equations differs only by the forcing terms. The derivation is therefore identical, except that the jumps across the domain are cancelled by finding an integral of the vorticity equation rather than using pre-existing constraint equations. The resulting layer equations are then

$$ {}^o\mathrm{KE}_{1t} - {}^o\rho \iint {}^o p_1 {}^o\eta_{1t}\, dA = {}^o\rho \iint {}^o p_1 {}^o e_1\, dA + {}^o\rho \iint ({}^o v_1 {}^o\tau^y + {}^o u_1 {}^o\tau^x)\, dA - {}^o\epsilon_{21} - {}^o\epsilon_{41}, \quad \text{(E.7)} $$

$$ \begin{aligned} {}^o\mathrm{KE}_{kt} + {}^o\rho \iint {}^o p_k\, ({}^o\eta_{k-1t} - {}^o\eta_{kt})\, dA &= \\ {}^o\rho \iint {}^o p_k\, ({}^o e_k - {}^o e_{k-1})\, dA - {}^o\epsilon_{2k} - {}^o\epsilon_{4k}, &\qquad k = 2, N-1, \quad \text{(E.8)} \end{aligned} $$

$$ \begin{aligned} {}^o\mathrm{KE}_{Nt} + {}^o\rho \iint {}^o p_N {}^o\eta_{N-1t}\, dA &= \\ - {}^o\rho \iint {}^o p_N {}^o e_{N-1}\, dA - \frac{{}^o\rho \delta_{ek} f_0}{2} \iint \left({}^o u_N{}^2 + {}^o v_N{}^2\right) dA - {}^o\epsilon_{2N} - {}^o\epsilon_{4N}, &\quad \text{(E.9)} \end{aligned} $$

where we have an additional term in the layer N equation representing linear bottom drag. The total energy equation is then

$$ \begin{aligned} {}^o\mathrm{KE}_{1t} + {}^o\mathrm{KE}_{2t} + \ldots + {}^o\mathrm{KE}_{Nt} + {}^o\mathrm{PE}_{1t} + \ldots + {}^o\mathrm{PE}_{N-1t} &= \\ - {}^o\rho \iint \left({}^o g'_1 {}^o\eta_1 {}^o e_1 + \ldots + {}^o g'_{N-1} {}^o\eta_{N-1} {}^o e_{N-1}\right) dA + {}^o\rho \iint ({}^o v_1 {}^o\tau^y + {}^o u_1 {}^o\tau^x)\, dA & \\ - \frac{{}^o\rho \delta_{ek} f_0}{2} \iint \left({}^o u_N{}^2 + {}^o v_N{}^2\right) dA - {}^o\epsilon_{21} - {}^o\epsilon_{41} - {}^o\epsilon_{22} - {}^o\epsilon_{42} - \ldots - {}^o\epsilon_{2N} - {}^o\epsilon_{4N}. &\quad \text{(E.10)} \end{aligned} $$

This equation shows buoyancy forcing, wind stress forcing, bottom drag and dissipation respectively on the right hand side. Each of these terms is independently recorded as a function of time in the monitoring section of the code.

References

Arakawa, A. and Lamb, V. R. (1977). Computational design of the basic dynamical processes of the UCLA general circulation model. *Meth. Comp. Phys.*, 17:173–365.

Arakawa, A. and Lamb, V. R. (1981). A potential enstrophy and energy conserving scheme for the shallow water equations. *Mon. Weather Rev.*, 109:18–36.

Golub, G. H. and van Loan, C. F. (1983). *Matrix Computations*. John Hopkins University Press, Baltimore.

Haidvogel, D. B., McWilliams, J. C., and Gent, P. R. (1992). Boundary current separation in a quasigeostrophic, eddy-resolving ocean circulation model. *J. Phys. Oceanogr.*, 22:882–902.

Holland, W. (1978). The role of mesoscale eddies in the general circulation of the ocean: numerical experiments using a wind-driven quasi-geostrophic model. *J. Phys. Oceanogr.*, 8:363–392.

Kravtsov, S. and Robertson, A. W. (2002). Midlatitude ocean-atmosphere interaction in an idealized coupled model. *Climate Dyn.*, 19:693–711.

McDougall, T. and Dewar, W. (1998). Vertical mixing and cabbeling in layered models. *J. Phys Oceanogr.*, 28:1458–1480.

McWilliams, J. C. (1977). A note on a consistent quasigeostrophic model in a multiply connected domain. *Dyn. Atmos. Oceans*, 1:427–441.

Opsteegh, J. D., Haarsma, R. J., Selten, F. M., and Kattenberg, A. (1998). ECBilt: a dynamic alternative to mixed boundary conditions in ocean models. *Tellus*, 50A:348–367.

Pedlosky, J. (1987). *Geophysical Fluid Dynamics*. Springer-Verlag.

Peixoto, J. P. and Oort, A. H. (1992). *Physics of Climate*. Springer-Verlag.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in Fortran 77: The Art of Scientific Computing (2nd ed.)*. Cambridge University Press.