The barotropic vorticity equation

The *barotropic vorticity equation* describes the evolution of a homogeneous (constant density), non-divergent, incompressible flow on the surface of the sphere.

For a homogeneous fluid in the absence of any non-conservative forces such as friction, Kelvin's circulation theorem tells us that the *circulation*, the line integral of the flow around any *material* loop, is conserved in time. Using Stokes' Theorem, one can translate this into the statement that the surface integral over any surface bounded by this loop, of the component of the vorticity normal to the surface, is conserved. For an infinitesimal loop constrained to move on the surface of the sphere, with area δA, this reduces to the conservation, following the flow, of $\omega\delta A$, the radial component of the vorticity times the area of the loop. If the flow is non-divergent, δA is conserved, so the radial component of the vorticity is itself conserved following the flow:

$$\frac{D\omega}{Dt} = 0, \tag{1}$$

or

$$\frac{\partial\omega}{\partial t} = -\mathbf{v}\cdot\nabla\omega = -\nabla(\mathbf{v}\omega) \tag{2}$$

We can define a streamfunction ψ as the unique solution to Poisson's equation, $\nabla^2\psi = \zeta$, on the surface of the sphere. If the flow is nondivergent along the surface, then the vorticity or the streamfunction define the flow completely through $\mathbf{v} = \mathbf{k}\times\nabla\psi$, where $\mathbf{k}$ is a unit vector in the radial direction. Therefore, the vorticity equation provides a self-contained equation of motion for this flow.

1 Spherical coordinates

Longitude λ ranges from 0 to 2π, and latitude θ from $-\pi/2$ at the South Pole to $\pi/2$ at the North Pole. Let u be the zonal (eastward) velocity and v be the northward velocity at constant radius. The divergence D and radial component of the vorticity ω on the surface of a sphere of radius a take the form

$$D \equiv \frac{1}{a\cos(\theta)}\frac{\partial u}{\partial\lambda} + \frac{1}{a\cos(\theta)}\frac{\partial}{\partial\theta}\Big(v\cos(\theta)\Big) \tag{3}$$

$$\omega \equiv \frac{1}{a\cos(\theta)}\frac{\partial v}{\partial \lambda} - \frac{1}{a\cos(\theta)}\frac{\partial}{\partial \theta}\Big(u\cos(\theta)\Big) \tag{4}$$

The relation between vorticity and streamfunction is

$$\nabla^2\psi = \frac{1}{a^2\cos^2(\theta)}\frac{\partial^2\psi}{\partial\lambda^2} + \frac{1}{a^2\cos(\theta)}\frac{\partial}{\partial\theta}\Big(\cos(\theta)\frac{\partial\psi}{\partial\theta}\Big) = \omega \tag{5}$$

and the flow field can then be reconstituted from ψ using

$$u = -\frac{1}{a}\frac{\partial\psi}{\partial\theta} \tag{6}$$

$$v = \frac{1}{a\cos(\theta)}\frac{\partial\psi}{\partial\lambda} \tag{7}$$

The material derivative for an incompressible flow can be written as

$$\frac{D\xi}{Dt} = \frac{\partial\xi}{\partial t} + \mathbf{v}\cdot\nabla\xi = \frac{\partial\xi}{\partial t} + \nabla\cdot(\mathbf{v}\xi) = \frac{\partial\xi}{\partial t} + \frac{u}{a\cos(\theta)}\frac{\partial\xi}{\partial\lambda} + \frac{v}{a}\frac{\partial\xi}{\partial\theta} \tag{8}$$

or

$$\frac{D\xi}{Dt} = \frac{\partial\xi}{\partial t} + J(\psi,\xi) \tag{9}$$

where the Jacobian, J, is defined as

$$J(A,B) = \frac{1}{a^2\cos(\theta)}\Big(\frac{\partial A}{\partial\lambda}\frac{\partial B}{\partial\theta} - \frac{\partial B}{\partial\lambda}\frac{\partial A}{\partial\theta}\Big) \tag{10}$$

If we view the flow from a rotating system with angular velocity Ω, and interpret (u,v) as the flow as observed in this rotating frame, the only point where the rotating frame is apparent is that the total (or *absolute*) vorticity of the flow now consists of two parts, the vorticity of solid body rotation, $f \equiv 2\Omega\sin(\theta)$, and the *relative* vorticity, the radial component of the curl of (u,v): $\omega = f + \zeta = f + \nabla^2\psi$. As f is independent of time, the material derivative of the absolute vorticity is simply $D\zeta/Dt$.

For the barotropic vorticity equation, we have

$$\frac{\partial\zeta}{\partial t} = -J(\psi, f+\zeta) = -\beta v - J(\psi,\zeta) = -\frac{2\Omega}{a^2}\frac{\partial\psi}{\partial\lambda} - J(\psi,\zeta) \tag{11}$$

where $\beta \equiv 2\Omega\cos(\theta)/a$ is the meridional gradient of f, and we are now using the symbols u, v, and ψ for the flow in the rotating coordinate system.

Since the integrals over the sphere of $AJ(A,b)$ and $BJ(A,b)$ vanish identically, the integrals over the sphere of $\zeta\partial\zeta/\partial t$ and $\psi\partial\zeta/\partial t$ vanish identically for this flow. The first of these is proportional to the rate of change of enstrophy, $\zeta^2/2$, and second to the rate of change of the kinetic energy $(u^2+v^2)/2$ (as one can see by an integration by parts). Therefore, both kinetic energy and enstrophy are conserved in time. All higher moments of vorticity are conserved as well, but enstrophy conservation is of special importance in that it is quadratic in the flow, and the existence of two quadratic invariants constrains the flow in important ways.

Solutions of this equation are known to be well-posed, in that they do not form a singularity in finite time if one starts with smooth initial conditions. So if one had infinite resolution, there would be no need for dissipation. But nonlinear, turbulent solutions in this two-dimensional geometry cascade enstrophy to smaller and smaller scales. In a finite resolution model, the enstrophy must be dissipated near the smallest resolved scales to simulate the loss of this vorticity variance to still smaller scales. In spectral models, one easily implemented solution is to add a hyperdiffusion of vorticity of the form:

$$\frac{\partial\zeta}{\partial t} = -J(\psi, f+\zeta) - \nu(-1)^n\nabla^{2n}\zeta \tag{12}$$

Hyperdiffusion of this sort is used because ordinary linear diffusion is often too dissipative for many applications. However, one should keep in mind that ordinary diffusion has no particular standing in this context either. We are interested not in a truly two-dimensional fluid but in planetary scale flows that are two-dimensional at large scales. There is a large spectral range, within which these equations are not valid, connecting the large scales of interest to the Kolmogorov microscale at which molecular diffusion would come into play. In any case, there is no theory underlying hyperdiffusion, and it is often criticized as ignoring the "backscatter" from small to large scales, etc.

An enstrophy cascade is characterized by a rate of transfer of enstrophy, ϵ, down the spectrum from large to small scales. The units of ϵ are $\zeta^2/T = T^{-3}$ where T is a time scale. So the strength of the cascade is simply characterized by a time scale. (This is unlike three-dimensional turbulence in which the time scale of the eddies decreases with the scale of the eddies.) Based on this heuristic picture, as one varies the resolution one often modifies ν so as to maintain the same damping time scale for the smallest resolved scale in the model, but there is no guarantee that this is always the most appropriate way of changing the diffusion as the resolution of the model is altered.

2 Time differencing

The code uses a standard leapfrog scheme with a time filter to control time-splitting. The time step is

$$\frac{\zeta_{i+1} - \zeta_{i-1}}{2\Delta t} = -J(\psi_i, f + \zeta_i) - \nu(-1)^n \nabla^{2n} \zeta_{i+1} \tag{13}$$

followed by the filter

$$\zeta_i = (1 - 2r)\zeta_i + r(\zeta_{i+1} + \zeta_{i-1}) \tag{14}$$

where r is a small number. This is sometimes referred to as a Robert filter, and sometimes as an Asselin filter. The damping is treated implicitly, indeed it is fully backward, with the vorticity within the damping term evaluated at ζ_{i+1}, so that it does not create artificial oscillations or growth. In the absence of damping, and with $r = 0$, the quadratic conserved quantities are now the integrals over the sphere of the staggered-in-time products $\zeta_{i-1}\zeta_i$ and $\psi_{i-1}\zeta_i$.

3 Spherical harmonic transform method

The equations are solved by writing the vorticity as a sum of spherical harmonics:

$$\zeta(\theta, \lambda) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \zeta_{lm} Y_{\ell m}(\lambda, \theta) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \zeta_{lm} P_{\ell m}(\sin(\theta)) e^{im\lambda} \tag{15}$$

Here m is the zonal, or azimuthal, wavenumber and ℓ is the total horizontal wavenumber, in the sense that $Y_{\ell m}$ is an eigenfunction of the Laplacian on the unit sphere with eigenvalue $-\ell(\ell+1)$. The $P_{\ell m}$ are the associated Legendre polynomials. If we define $n \equiv \ell - m$, then n is the number of zero's in latitude, with even (odd) n resulting in a function that is even (odd) about the equator. To insure that ζ is real, we require that $\zeta_{\ell -m} = \zeta^*_{\ell m}$, where the asterisk denotes complex conjugation. We can also write the sum as

$$\zeta(\theta, \lambda) = \sum_{\ell=0}^{\infty} \zeta_{\ell 0} P_{\ell 0}(\sin(\theta)) + 2 \sum_{\ell=0}^{\infty} \sum_{m=1}^{\ell} P_{\ell m}(\sin(\theta)) \Re\left(\zeta_{\ell m} e^{im\lambda}\right) \tag{16}$$

where $\Re$ refers to the real part. Here and in the code the spherical harmonics are normalized such that

$$\frac{1}{2\pi}\int_{-\pi/2}^{\pi/2}\cos(\theta)d\theta\int_{0}^{2\pi}d\lambda Y_{\ell m}(\lambda,\theta)Y^{*}_{\ell' m'}(\lambda,\theta)=\delta_{mm'}\delta_{\ell\ell'} \tag{17}$$

Note that this implies that, for example, $Y_{00}=1/\sqrt{2}$, that is, the spherical harmonics are normalized so that their squared amplitude integrates to unity over *one hemisphere*.

In a *triangular truncation*, the most popular, one truncates this spectrum at some value of $\ell = L$, and then retains all values of m for each ℓ in the truncation. The truncation is rotationally symmetric, in the sense that if one rotates a function that is expressible within this truncation, then the rotated function remains within this truncation. (More generally, rotation does not mix different ℓ's). Less often in recent years, one also sees *rhomboidal truncation*, in which one retains the same number of meridional modes for each zonal mode, or, equivalently, all modes with $n = \ell - m$ less than or equal to some N. The term rhomboidal truncation is usually restricted to the case in which $N = M$, so that the number of zonal model is equal to the number of meridional modes, but other choices are occasionally made. Rhomboidal truncation is not rotationally symmetric.

One can write equations for the time evolution of the spectral coefficients in term of the spectral coefficients themselves. The nonlinear advection term takes one outside of the truncation; one defines the spectral model by retaining only that part of the nonlinear term that projects onto the retained truncation, a procedure that produces a truncated model that conserves energy and enstrophy. The expression for the advection term involves multiple convolution summations, and, as a consequence, the resulting *pure* spectral model is hopelessly inefficient for even modest truncations. The efficient use of spectral representations in models such as this depends on the *spectral transform* method in which one returns to physical space where needed, especially in order to perform multiplications. If one returns to physical space at the appropriate resolution, choosing the $de-aliasing$ grid appropriate for one's choice of truncation, one can reproduce the truncated pure spectral model exactly, conserving energy and enstrophy as before.

The grid that one returns to is equally spaced in longitude but not equally spaced in latitude; rather the meridional grid points are given by the Gaussian latitudes, which are resolution dependent. The Gaussian grid is symmetric about the equator. For a discussion of Gaussian quadrature in the context of spectral models, as well as de-aliasing grids, see

Durran, D. R., 1999: *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics*, Springer Verlag, New York.

Consider a Fourier series in one dimension, truncated at wavenumber M. If we evaluate this function at N equally spaced points, will we be able to reconstruct the exact Fourier spectral representation with a discrete Fourier transform? The answer is yes if $N \geq 2M + 1$. One can remember this by noting that a real field represented by a Fourier series truncated at $m = M$ has $2M + 1$ real degrees of freedom (the $m = 0$ spectral amplitude is real, while $m = 1, M$ are complex) so one could not possibly go back and forth with fewer grid points without losing information.

Now suppose that one takes two such truncated functions and multiplies them together. Since this product contains wavenumbers that are twice as large as the individual factors, one would need $4M + 1$ grid points if one wanted to move the product to grid space and then back to the spectral domain retaining the exact expression for the product. Equivalently, this number of grid points would allow one to move each component to the grid, multiply them together on the grid, and then transform back to the spectral domain, obtaining the exact answer. But in a spectral method for an equation with a quadratic nonlinearity, this is more than one needs. One requires only the projection of the product onto the original truncation (that is, the first M wavenumbers). The exact computation of this component of the product requires only $3M + 1$ grid points. It is this grid that is referred to as the de-aliasing grid for quadratic products. One can think in terms of the quadrature that one must perform exactly to compute a spectral component of some function f – one multiplies f by a trigonometric function within one's truncation and then integrates over the domain. In this case f is a product containing wavenumbers up to $2M$ and the multiplier needed to project out the required spectral component can range up to wavenumber M, so the integrand will have wavenumbers up to $3M$. To evaluate this integral exactly requires $3M + 1$ points. (Admittedly, it is hard to understand the "+1's" in this casual way.)

Now define a field on the sphere with triangular truncation at wavenumber M and ask the same questions. It turns out that to move this function to a grid and back without loss of information requires at least $2M + 1$ equally spaced points in longitude and a Gaussian grid with at least $M+1$ points. This is sometimes referred to as the *linear* grid. Note however, that one cannot define an arbitrary function on this grid, move it to the spectral domain and then back to the grid, and retain the same grid values (unlike the Fourier case). One can understand this by noting that the transform grid is a latitude-longitude grid, with points along a latitude circle closer and closer together as one approaches the pole. Clearly this grid is not spherically isotropic, and an arbitrary function defined on this grid cannot be represented by a triangularly truncated function, which can have no distinguished pole in the sense described earlier. If one makes the transformation $grid \rightarrow spectral \rightarrow grid_1 \rightarrow spectral \rightarrow grid_2$ then $grid_2 = grid_1$, for then $grid_1$ has been filtered to contain only the harmonics within the truncation.

For a de-aliasing grid for quadratic products, one finds that the requirements are at least $3M+1$ longitudes and $(3M+1)/2$ meridional Gaussian latitudes. For rhomboidal truncation at zonal wavenumber M (with $N = M$), the de-aliasing grid requires $3M+1$ longitudes and $(5M+1)/2$ latitudes.

One can compute u and v from the vorticity in the spectral domain by first computing the streamfunction (by simply dividing by the appropriate eigenvalue of the Laplacian). Then $v\cos(\theta)$ can be obtained immediately since a zonal derivative consists of a simple multiplication by im in the spectral domain. The meridional derivative needed to compute u can be obtained with a recursion relation. More precisely, $u_{\ell m}\cos(\theta)$ is a linear combination of $\psi_{\ell+1,m}$ and $\psi_{\ell-1,m}$. The underlying recursion relation is

$$\cos(\theta)\frac{dP_{\ell m}}{d\theta} = -\ell\epsilon_{\ell+1,m}P_{\ell+1,m} + (\ell+1)\epsilon_{\ell,m}P_{\ell-1,m} \tag{18}$$

where

$$\epsilon_{\ell,m} \equiv \left(\frac{\ell^2 - m^2}{4\ell^2 - 1}\right)^{1/2} \tag{19}$$

Starting with the spectral expansion of ψ, differentiating with respect to θ and multiplying by $\cos(\theta)$, using this recursion relation and then collecting terms proportional to $P_{\ell m}$, we find that

$$(\cos(\theta)u)_{\ell m} = -\frac{1}{a}(-(\ell-1)\epsilon_{\ell,m}\psi_{\ell-1,m} + (\ell+2)\epsilon_{\ell+1,m}\psi_{\ell+1,m}) \tag{20}$$

So as not to lose information when one computes this derivative, one must retain one more meridional mode, for each m, in the truncation of $u\cos(\theta)$

than for ζ or ψ. (In the code, for convenience, all spectral fields are dimensioned so as to allow for this extra harmonic, with the understanding that scalars such as ζ will not make use of the last meridional mode.)

One can also compute vorticity and divergence from u and v, but now the recursion relation requires the spectral components of $u/\cos(\theta)$ and $v/\cos(\theta)$. To see this, note that to compute $\zeta_{\ell m}$ we have to compute an integral of the form

$$\int P_{\ell m} \frac{1}{\cos(\theta)} \frac{\partial}{\partial \theta}\Big(u\cos(\theta)\Big)\cos(\theta)d\theta = -\int \cos(\theta)\frac{\partial P_{\ell m}}{\partial \theta}\left(\frac{u}{\cos(\theta)}\right)\cos(\theta)d\theta \tag{21}$$

One can then use the recursion relation (18) for $\cos(\theta)\partial P_{\ell m}/\partial\theta$ to obtain $\zeta_{\ell m}$ from the spectral decomposition of $u/\cos$ (and $v/\cos$). Starting with vorticity and divergence (the latter equal to zero in this nondivergent model), one can compute $u\cos(\theta)$ and $v\cos(\theta)$ in the spectral domain, move these to the grid domain, divide by $\cos^2(\theta)$, then compute the spectral components of vorticity and divergence, returning to exactly the same expressions as one started with (as long as the grid is at least as large as the linear grid and as long as one is careful to retain the additional meridional mode for $u\cos$ and $v\cos$, as well as $u/\cos$, $v/\cos$).

4 Algorithm

An outline of the steps involved in integrating the nondivergent barotropic spectral model is as follows. Assume that we know both the spectral and grid vorticity, and the grid values of u and v, at $t-\Delta t$ and t. Then

1: Add the Coriolis force to the grid relative vorticity at time t

2: compute $(f+\zeta)u$ and $(f+\zeta)v$ on the grid at time t – think of $\partial u/\partial t = (f+\zeta)v$ and $\partial v/\partial t = -(f+\zeta)u$

3: compute the spectral divergence of $((f+\zeta)u, (f+\zeta)v)$, or, equivalently, the spectral curl of $((f+\zeta)v, -(f+\zeta)u)$, to obtain the vorticity tendency due to advection, Z, by first dividing these tendencies by $cos(\theta)$, transforming to the spectral domain and using the recursion relation mentioned in the previous paragraph.

4: add the biharmonic damping to this advective tendency in the spectral domain, treating the damping implicitly,

$$Z \to \frac{Z - \nu(2\Delta t)\sigma^n \tilde{\zeta}}{1 + \nu(2\Delta t)\sigma^n} \tag{22}$$

where σ is the (absolute value of the) eigenvalue of the Laplacian for each spherical harmonic,

5: use leapfrog to generate the spectral vorticity $\zeta(t + \Delta t)$ and apply the Robert filter to modify the spectral vorticity at time t.

6: compute grid relative vorticity, as well as the spectral $u \cos$ and $v \cos$ and then grid u and v, at $t + \Delta t$ from the spectral vorticity.

Note that we do not bother to compute the new grid vorticity at time t resulting from the Robert filter. This extra transforms does not seem to be needed to maintain smooth temporal evolution in these barotropic simulations, since the memory resides in the spectral domain.

There is an option of carrying a passive scalar tracer, using spectral advection, just as for the vorticity. For development purposes, there is also an option of carrying along another passive tracer which is a grid-point variable advected using a piecewise linear finite volume technique. (A piecewise parabolic version is in the works, as is a version in which the vorticity itself is advected with the finite volume scheme).

The finite volume advection scheme follows closely that described in

Lin, S.-J. and R. B. Rood, 1996: Multidimensional flux-form semi-Lagrangian transport schemes. Monthly Weather Review, 124, pp. 2046-2069.

and is discussed further in the documentation for the module *fv_advection_mod*. This scheme is not particularly suited for use in a leapfrog context. The procedure used here carries two time levels of information for the tracer ξ, just as for vorticity. Using the velocities at time t, we call the finite volume advection algorithm to advect the scalar ξ from $t-\Delta t$ to $t+\Delta t$; then we Robert filter ξ. There is no explicit diffusion of tracer when using the finite-volume scheme. The integrations are started with a simple forward step of length Δt.

5 Default example

By default the programs runs an example of the decay of a sinusoidal disturbance to a zonally symmetric flow that resembles that found in the upper troposphere in Northern winter. We start with the zonal flow

$$u = 25\cos(\theta) - 30\cos^3(\theta) + 300\sin^2(\theta)cos^6(\theta) \tag{23}$$

and modify the resulting vorticity field by adding a perturbation of the form

$$\zeta' = \frac{A}{2}\cos(\theta)e^{-((\theta-\theta_0)/\theta_W)^2}cos(m\lambda) \tag{24}$$

The default choices of $m = 4$, $\theta_0 = 45°N$, $\theta_W = 15°$, and $A = 8.0 \times 10^{-5}$ produce an interesting evolution art high resolution. (All units are MKS, i.e., u is in m/s and ζ in $1/s$). The radius of the sphere is set to the radius of the Earth, as defined in *constants_mod*. If the spectral or grid tracers, or both, are turned on, they are, by default initialized to equal to $+1$ in the zonal strip between $10°N$ and $20°N$, and equal to -1 north of $70°N$.

This initial value problem has been studied in detail, in the linear limit, ($A \to 0$), by

Held, I., 1985: Pseudomomentum and the orthogonality of modes in shear flows. Journal of the Atmospheric Sciences, 42(21), 2280-2288.

and in the nonlinear case by

Held, I. M. and P. J. Phillips, 1987: Linear and nonlinear barotropic decay on the sphere. Journal of the Atmospheric Sciences, 44(1), 200-207.

In choosing the resolution, the number of zonal grid points, num_lon, should be factorisable into powers of 2,3 and 5 so as to utilize the fast Fourier transform.

To set the resolution, you need to define num_lon, num_lat, num_fourier, and num_spherical. For the triangular model TM (i.e. T21 with $M = 21$) with a de-aliasing grid, we always have num_lon = 2(num_lat), num_fourier = M, num_spherical = $M + 1$, so we need only determine M and num_lat $\equiv NY$. Popular choices include

M	NY	factors
21	32	2^5
31	48	$2^4 \times 3$
42	64	2^6
63	96	$2^5 \times 3$
85	128	2^7
106	160	$2^5 \times 5$
127	192	$2^6 \times 3$
170	256	2^8
213	320	$2^6 \times 5$
255	384	$2^7 \times 3$
341	512	2^9
511	768	$2^7 \times 5$
682	1024	2^{10}
1365	2048	2^{11}

The default for the diffusivity is $\nabla^8 = (\nabla^2)^4$ (*damping_order = 4*), with *damping_option = 1* and *damping_coeff = 1.e-04*. With this *damping_option*, the value of *damping_coeff* is the damping rate (in 1/s) for the highest meridional mode with $m = 0$. For triangular truncation this is equivalent to setting the damping rate for the largest value of ℓ within the truncation. If you instead set *damping_option = 2*, ν is simply set equal to the the input value of *damping_coeff*. (For higher resolutions with this test case, T341 and above, it seems that the strength of the diffusivity needs to be increased for a stable integration – 1.e-03 seems to be safe in all cases with ∇^8.)

The time step also must be adjusted to the resolution. For T85, $\Delta t = 1800s$ is fine. Decreasing the time step proportionally should work as the resolution is increased.

Since the initial condition is periodic in λ with wavenumber m in this case, one can save computations by integration on a $360/m$ degree sector by setting fourier_inc = m and by dividing both num_fourier and num_lon by m, taking care to retain an appropriate factorization for num_lon. For example, for $m = 4$ one can try fourier_inc = 4, num_fourier = 42, num_spherical = 171, num_lon = 128, num_lat = 256.

The transforms module is currently restricted to 1D domain decomposition, using latitude in the grid domain and fourier waves in the spectral domain. Furthermore, it requires the same even number of latitudes on each processor. So the number of processors must divide evenly into num_lat/2.

6 Structure of the code

There is a generic main program, in *main.f90*, that is used in a number of idealized atmospheric models. Besides doing some bookkeeping, it includes the main time loop and has a namelist in which the time step Δt and the length of the integration are provided.

The main program runs the model by calling the routines in *atmosphere_mod*, which, in turn, use routines in *barotropic_dynamics_mod*, *barotropic_physics_mod*, *barotropic_diagnostics_mod*, and *fv_advection_mod*. *barotropic_physics* does nothing in the default version of the code. Model resolution, the spectral damping, and the strength of the time filter, r, are controlled by a namelist read by *barotropic_dynamics_mod*.

The dynamics module uses *transforms_mod*, which contains a variety of routines for transforming data from spherical harmonics to a grid and back, computing derivatives in the spectral domain, etc.

Diagnostics are controlled by the diagnostics manager. Those immediately available are $u, v, \zeta, f + \zeta, \psi, \xi_s, \xi_g$ where the latter two are the gridded tracer fields generated, respectively, using spectral advection and finite-volume advection. All fields are output into netcdf files, one per processor, that can automatically be combined into full spatial fields using FMS's *mpp-nccombine* utility. The time interval at which output is generated, file names, etc ate controlled from the *diag_table* file read by the diagnostics manager. To add additional diagnostic fields, follow the template in *barotropic_diagnostics_mod*.

Restart files are generated by *atmos_model_mod* and *barotropic_dynamics_mod*. The former contains information about time, while the latter contains the state of the model. If dir is the directory in which the code is being executed, restarts are placed in dir/RESTART (which must exist). If these files are copied to dir/INPUT then the model will read them and continue the integration smoothly if started up again. Namelists must also reside in dir/INPUT.