In this project, I implement the barotropic vorticity equation, which is the standard building block for constructing more realistic General Circulation Models (GCMS) in atmosphere and ocean science, on the 2D surface of the rotational sphere. The equation is solved on spherical coordinate, so it is natural to take the spherical harmonics as the basis for solving the system with pseudo-spectral method. Due to the complexity of the spherical basis (especially as the resolution of the problem increases) and lack of efficient schemes for fast transform, the most time consuming part of this method happens at the frequent transforms between physical and spectral domain during the calculation at each time instant. On the other hand, the properties of the spherical harmonics make it convenient and easy to be implemented with MPI and OpenMP. The major part of this project is to focus on accelerating the the transforms between domains through parallel.

1. The Barotropic Vorticity Equation and Its Numerical Discretization

1.1. 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 that the circulation is conserved in time. Particularly, for 2D non-divergent flow the (absolute) vorticity is itself conserved following the flow

\[
\frac{\partial \omega}{\partial t} = -\nabla \cdot \omega = -\nabla (\omega \cdot \mathbf{v}).
\]

We can define the streamfunction \(\psi\) as the unique solution to Poisson’s equation, \(\omega = \nabla^2 \psi\), 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.

If we view the flow from a rotating system with angular velocity \(\Omega\), 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 absolute (or total) vorticity \(\omega\) of the flow now consists of two parts, the vorticity of solid body rotation, \(f = 2\Omega \sin(\theta)\), and the relative vorticity of the curl of \((u, v)\), that is, \(\omega = f + \zeta = f + \nabla^2 \psi\). Therefore the barotropic flow (1.1) on the rotational sphere becomes

\[
\frac{\partial \zeta}{\partial t} = -J (\psi, f + \zeta) - \nu (-1)^m \nabla^{2m} \zeta,
\]

with the relative vorticity \(\zeta = \nabla^2 \psi = \nabla \times \mathbf{v}\) and \(J\) defined as the Jacobian. Note that we add an additional term \(-\nu (-1)^m \nabla^{2m} \zeta\) on the right hand side of the equation representing the hyperdiffusion. This is an artificial term to dissipate the unresolved small scale energy in finite resolution spectral models. There is no theory underlying this hyperdiffusion, and the diffusion rate \(\nu\) and diffusion order \(m\) are both chosen empirically depending on the model resolution.

1.2. Numerical discretization. We follow the basic method-of-lines process to solve this equation. Specifically, use pseudo-spectral method to compute the spatial discretization, and use standard
leapfrog scheme for time updating, then the numerical scheme is like
\[
\frac{\zeta_{n+1} -\zeta_{n-1}}{2\Delta t} = -J (\psi_n, f + \zeta_n) - \nu (-1)^m \nabla^2 m \zeta_{n+1}.
\] (1.3)

The second order leapfrog scheme is generally used in climate models, but suffers numerical dispersion and artificial oscillations. To eliminate these undesirable effects, a Robert filter (or sometimes as an Asselin filter) is added
\[
\zeta_n = (1 - 2r) \zeta_n + r (\zeta_{n+1} + \zeta_{n-1}),
\] (1.4)

where \( r \) is chosen as a small number. The hyperdiffusion part \( \nabla^2 m \zeta_{n+1} \) is usually stiff and thus is treated implicitly. It serves as additional damping that dissipates the energy cascaded to unresolved smaller scales.

Now the final issue is about the calculation about the spatial differentiation in the Jacobian \( J \). Variables on the sphere for spectral method is decomposed under the spherical harmonic basis, for example for the vorticity \( \zeta \)
\[
\zeta (\lambda, \mu) = \sum_{m=-\infty}^{\infty} \sum_{n=|m|}^\infty \hat{\zeta}_{m,n} Y_{m,n} (\lambda, \mu) = \sum_{m=-\infty}^{\infty} \sum_{n=|m|}^\infty \hat{\zeta}_{m,n} P_{m,n} (\mu) e^{im\lambda}.
\] (1.5)

with \( \lambda \in [0, 2\pi) \) the longitude, \( \mu = \sin \theta, \theta \in [-\pi/2, \pi/2] \) the latitude. \( Y_{m,n} \) is the eigenfunction for the Laplacian operator \( \nabla^2 \) on the sphere with corresponding eigenvalue \( \sigma_{m,n} = -\frac{n(n+1)}{L^2} \), and \( P_{m,n} \) is the associated Legendre functions. Following these conventions, the equation can be written for the time evolution of the spectral coefficients \( \hat{\zeta}_{m,n} \) in terms of the spectral coefficients themselves. Derivative about the longitude \( \lambda \) then becomes a multiplication of \( im \) from the Fourier decomposition, and derivative about the latitude \( \theta \) can be achieved through the recursive formula for Legendre functions
\[
\cos \theta \frac{dP_{m,n}}{d\theta} = -n \epsilon_{m,n+1} P_{m,n+1} + (n+1) \epsilon_{m,n} P_{m,n-1},
\] (1.6)

where \( \epsilon_{m,n} = \left( \frac{n^2 - m^2}{4n^2 - 1} \right)^{1/2} \). Therefore the derivatives can be converted to multiplication of eigenvalues and simple applications of the recursive relation above.

In all practical application, the infinite series (1.5) must be truncated to create a numerical approximation of the form
\[
a (\lambda, \mu) = \sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} a_{m,n} P_{m,n} (\mu) e^{im\lambda}.
\]

Here \( M \) the largest truncated Fourier mode in zonal direction \( \lambda \), and \( P_{m,n} \) is only non-zero when \( n \geq |m| \) for any chosen \( |m| \leq M \). Different truncation methods for the largest wavenumber \( N (m) \) in meridional direction are suggested. The most popular one is the triangular truncation, in which \( N (m) = M \), is unique among the various possible truncations because it is the only one that provides uniform spatial resolution over the entire surface of the sphere. This truncation also has the desirable property of invariance under arbitrary rotation of the latitude and longitude about the center of the sphere. Another common alternative is the rhomboidal truncation, in which \( N (m) = |m| + M \). This truncation retains the same number of meridional modes \( n \) for different zonal wavenumber \( m \). But the disadvantage is that this truncation loses its symmetry under rotations.

One additional issue for spectral methods is the arrangement for nonlinear terms (such as the Jacobian in the vorticity equation above). Dealiasing for high wavenumber truncation is required to avoid aliasing errors from mixing the unresolved high wavenumber modes with the resolved low wavenumber modes. Simply we apply the standard 2/3-rule introducing \( 3M + 1 \) grid points in the \( 2M + 1 \) discretization.
2. Transforms between Physical and Spectral Domain in Parallel

Here we discuss the parallel approach to accelerate the code. As already noted above, the most time consuming part in the pseudo-spectral codes happens at the frequent transforms of the state variables of interest between the physical domain and the spectral domain under spectral harmonics. For clarification, we decompose the spectral transform of some state variable \( a \) into two parts as follows:

\[
a(\lambda, \mu) = \sum_{m=-M}^{M} \hat{a}_m(\mu) e^{im\lambda}, \quad \hat{a}_m(\mu) = \sum_{n=|m|}^{N(m)} a_{m,n} P_{m,n}(\mu).
\]

The first part is just the standard discrete Fourier transform which can be efficiently implemented using the FFT package with complexity \( O(M \log M) \), whereas the no efficient implementation for the second part including associated Legendre polynomials \( P_{m,n} \) is available (there exists some faster algorithms for this transform using fast cosine transforms, but it seems that the improvements can only be obvious when the truncation \( M \) becomes extremely large). So the multiplication of Legendre polynomials requires complexity of order \( O(M^2) \). But on the other hand, note that i) the Fourier transforms on each latitude \( \mu \) is independent with each other; ii) in the recursive formula for calculating differentiations in (1.6), updates for each zonal wavenumber \( m \) are independent with each other only including varying modes in \( n \). These two desirable properties in the spherical harmonics make it natural to implement the transform in a parallel fashion. In the first Fourier transform in (2.1), different processors can take different latitude bands in range \( [\mu_1, \mu_2] \) and carry out FFT independent with each other. Then different processors communicate the information the calculated. In the second transform for Legendre polynomials in (2.1), each processor collect proper band for zonal modes in range \( [m_1, m_2] \), and again, they can calculate the transforms independently as well as the differentiations in spectral domain.

We summarize the parallel spherical transform algorithm as follows:

**Algorithm.** (transform from physical domain to spectral domain)

- Assign each processor a latitude band with equal size in range \( \mu \in [\mu_1, \mu_2] \) with no intersection with each other. And each processor can carry out standard 1D FFT along the latitude circle to calculate the zonal modes at \( \mu \):
  \[
  \hat{a}_m(\mu) = \frac{1}{2NY} \sum_{j=0}^{2NY-1} a(\lambda, \mu) e^{-im\lambda}.
  \]

- Communicate the zonal modes between processors. Specifically, each processor gets a band with equal size of zonal modes in range \( m \in [m_1, m_2] \) with no intersection with each other including all the latitude grids \( \mu \).

- After receiving all the required modes in the range \( [m_1, m_2] \) from other processors, calculate the spectral modes using Gaussian quadratures
  \[
  a_{m,n} = \int_{-1}^{1} \hat{a}_m(\mu) P_{m,n}(\mu) d\mu = \sum_{j=0}^{NY-1} \hat{a}_m(\mu_j) P_{m,n}(\mu_j) w_j.
  \]

The process can also be carried out independently with each processor. And Gaussian quadrature abscissas are taken to make sure the accuracy of this integration.

Inversely, the inverse transform can be carried out in a similar fashion also independent with each processor during the calculations of the two parts of transforms and a simple communication of modes between them. Besides the multi-process parallel under MPI in each processor, it is also possible to parallel the computation under multi-thread computation inside the calculation in each process using OpenMP or GPU if the scale of the calculation is even larger.
3. Numerical Simulations

To test the performance of the method, we set the initial values starting with zonal flow
\[ u = 25 \cos \theta - 30 \cos^3 \theta + 300 \sin^2 \theta \cos^6 \theta, \quad v = 0, \]
and modify the resulting vorticity field by adding a perturbation
\[ \zeta' = \frac{A}{2} \cos \theta e^{-((\theta-\theta_0)/\theta_W)^2} \cos (m\lambda). \]
In this test case, we choose relatively coarse resolution with \( NY = 256, M = 170 \). (For the purpose of FFT, the meridional resolution \( NY \) is better to be chosen as the orders of prime numbers, that is \( NY = 2^\alpha \cdot 3^\beta \cdot 5^\gamma \); and the number of processors must be factors of \( NY \) and \( M \) for even partition of modes.) The initial fields are plotted in Figure 3.1 for both the velocity and vorticity fields. Following, Figure 3.2 offers four snapshots of the vorticity field at different time \( t = 0, 200, 400, 800 \) as the flow evolves. For comparison, the results with irrotational sphere and a rotational sphere are both plotted. It can be seen that at a rotating earth, the large vortices quickly break up into smaller scales and finally dissipated at the unresolved level, whereas for a non-rotating earth, the large vortices keep their shape and get advected along the flow.

4. Possible Further Directions

I plan to finish the following steps in the future:

- Visualize the results using Paraview.
- Solve the rotational shallow water equations on the sphere using the above pseudo-spectral strategy. Furthermore, it will be interesting to try the multi-level shallow water system with thermodynamics considered.
- Add passive tracers into the system, and use finite volume scheme to solve the advection equations for tracers. It is interesting to see how the tracers are advected by the flow on the sphere.

References

2. Geophysical Fluid Dynamics Laboratory, The barotropic vorticity equation.
(a) irrotational sphere $t = 0$

(b) rotational sphere $t = 0$

(c) irrotational sphere $t = 200$

(d) rotational sphere $t = 200$

(e) irrotational sphere $t = 400$

(f) rotational sphere $t = 400$

(g) irrotational sphere $t = 800$

(h) rotational sphere $t = 800$

Figure 3.2. Vorticity field at time $t = 0, 200, 400, 600$.

(3) Konstantin Holoborodko, *Numerical Integration by Gauss-Legendre Quadrature Formulas of high orders.*

(4) John Burkardt, *Calculating associated Legendre polynomials.*