Tracer transport model (plume dispersion) and the spectral method

Spherical harmonics

(some useful properties)

$$\psi(\lambda,\mu)=\sum_n\sum_m \psi_n^m P_n^m(\mu)e^{-im\lambda}$$

$$\nabla^2\psi_n^m=-\frac{n(n+1)}{a^2}\psi_n^m$$

$$\frac{\partial\psi_n^m}{\partial\lambda}=im\psi_n^m$$

$$(1-\mu^2)\frac{\partial\psi_n^m}{\partial\mu}=-n\varepsilon_{n+1}^m\psi_{n+1}^m+(n-1)\varepsilon_n^m\psi_{n-1}^m$$

$$\varepsilon_n^m=\sqrt{\frac{n^2-m^2}{4n^2-1}}$$

Advection with spherical harmonics

$$\frac{\partial q_n^m}{\partial t} = -\vec{V} \cdot \nabla q = -\nabla \cdot \left(\vec{V} q\right)$$

Non-divergent!

$$\frac{\partial q_n^m}{\partial t} = -\alpha(F,G)_n^m$$

$$\alpha(F,G) = \frac{1}{\cos^2 \phi}\left(\frac{\partial F}{\partial \lambda} + \cos\phi \frac{\partial G}{\partial \phi}\right)$$

$$F = \frac{\cos\phi}{a} uq \qquad G = \frac{\cos\phi}{a} vq$$

Assignment

- What fraction of pollution emitted in Colorado in January is in the Southern Hemisphere by the end of June?

$$\frac{\partial q}{\partial t} = -\nabla \cdot \left(\vec{V} q\right) + S + k\nabla^2 q$$

S is a source flux.

Use a centered finite difference for time

Use spherical harmonics to represent the spatial structure.

As such, the spatial derivatives are known analytically

Use grid nlon=64, nlat = 32, ntrn=21, ltriang=.true.

Add a “source” as a known flux of tracer into the atmosphere from the grid point nearest Colorado.

Implementing model code

- What is the state variable?
  (a series of complex coefficients)

- Read wind on a grid
- Obtain tracer field on a grid
- Calculate non-linear fluxes on a grid
  (*uq* and *vq*)

Spherical harmonic synthesis

- Assign the surface fluxes (S)
- Convert fluxes to spectral form
- Compute flux divergence in spectral form
  (just the advection, as per last week)
- Time step the state variable (in spectral form)

Spherical harmonic analysis

Remember to output state every so often (say, 24 hours?)

Use as 0.5 hour time step. 48 steps per day.

Spherical.F90 module

Fortran modules contain data, subroutines…
```
use spherical
```
Initialize module:
```
subroutine sph_init(nlon,nlat,ntrn,ltriang)
```
Do analysis:
```
subroutine sph_anal(nlon,nlat,nlev,grid,ntrn,spec)
```
Do synthesis
```
subroutine sph_synth(ntrn,nlev,spec,nlon,nlat,grid)
```
Compute spectral “alpha” from grid values:
```
subroutine sph_anal_alpha(ntrn,nlev,nlev,aunit,amn,bunit,bmn,ntrn,alpha)
```

See example code in ~dcn/ATOC7500/week08

sphexample.f90	(initialize and transform, and Laplacian)
sphalphaeg.f90	(compute divergence with ‘alpha’)
realncep.f90	(read the NCEP data)

Needs a “module” Spherical.F90, and the fft99.F subroutines
Compile, e.g., f90 fft99.F Spherical.F90 sphexample.f90
Also, plotting routine tplot.pro

Tips for implementing model

1. Set up time stepping framework, and output file (perhaps copy this from last weeks code)
2. Set number of total steps to 1 for testing
3. Set some initial qg. convince yourself you can to forward and backward transforms – plot them.
4. Set some simple wind field (say solid body rotation)
5. Compute and transform “F” and “G” and try a few steps
6. Add diffusion *[aim to get here by the end of class]*

7. Add source term (on grid, convert to spectral)

- Try reading observed wind field
  (you will need to use horizontal diffusion to he useful results)

- Keep the model as simple as possible until you are convinced it works.

Discussion questions

- How did you choose the diffusion coefficient?
- How does the diffusion change your answer?
- How does the frequency at which you update the wind change your answer?
- What do you expect happens to the tracer distribution after some long integration time?