Discussion Feb 16th, 4pm (neuer Hörsaal)

This practical gives a numerical perspective on some aspects of the quasi-geostrophic approximation. We will use the Quasi-geostrophic barotropic vorticity with and without a forcing and the two-layer model to reflect on some of the theoretical findings derived during the semester.

We devote the final exercise on Feb 16^{th} to a discussion of your results. Please hand in your code and a small exposé consisting of some figures showing results from the model prior to this discussion. We expect a brief discussion / presentation of the exposé during this exercise.

1 Preliminaries - model infrastructure 4 P

You can use the model infrastructure provided for python that is provided on the whiteboard. (Alternatively, you may develop an infrastructure with similar capabilities in a programming/scripting language of your choice.)

Make sure you understand the following aspects before you get started to implement any other task of this assignment:

- How is the model controlled (how does it learn about its input / parameters / configurations?) ?
- Which classes exist and what is there purpose?
  1. ______________ – ______________
  2. ______________ – ______________
  3. ______________ – ______________
  4. ______________ – ______________
- What time integration scheme is used? In wich class and routine is it implemented?

Scheme ______________________

Class ______________________

Routine ______________________

- What kind of output is generated to the following streams/files

stdout ______________________

*.log ______________________

*.iter ______________________

*.nc ______________________

2 Time integration 4 P

Check the time integration scheme implemented in the class `INTEGRATOR`. Setting `rhs` [1] to `exponential` (a member of the class `integrator`) evaluates to the input, i.e. we solve the equation

$$\frac{dx}{dt} = x \Rightarrow \tilde{x}(t) = \frac{x_0}{e^{t_0}} e^t.$$

a) What is the expected result for $t_0 = 0$ and $x_0 = 1$ when integrating up to $t = 2$? Check for $\Delta t = 1$!

b) Varying the time step τ of the integration, check the order of the time integration scheme and produce a log-log plot of the error $x_{num} - \tilde{x}$ as a function of τ.

c) What is the order of the numerical error for the time integration scheme used?

d) Why does the numerical error reach a limit as $\tau \to 0$?

3 Rossby waves 14 P

For this task, you need to use the `rhs` dummy routine `rossby` and the section `Rossby` in `qg.ini`

a) Implement the barotropic vorticity equation

$$\left[\frac{\partial}{\partial t} + (U_j + {u'}_j)\frac{\partial}{\partial x_j}\right]\nabla^2\zeta - \beta v = 0$$

b) Investigate the behavior of waves for a `harmonic` initial perturbation! Can you prodcue standing waves? Give the parmaeter setting!

c) Investigate the behavior of a `singular` perturbation!

- What do you observe?
- How does the behavior differ from a harmonic initial condition?
- Can you explain this?

4 Two-layer model 18 P

a) In this task, we implement the two-layer model using a prognostic equation for ζ at the 250 and 750 millibar level (assuming $p_{\text{sfc}} = const. = 1000\,\text{hPa}$). Deviating from Holton's Eq. 8.15 and 8.16, we may obtain

$$\frac{\partial(\nabla^2\psi_m)}{\partial t} = -\left[\mathbf{u}_m\nabla\right]\nabla^2\psi_m - \left[\mathbf{u}_T\nabla\right]\nabla^2\psi_T - \beta\frac{\partial\psi_m}{\partial x} = 0$$

$$\frac{\partial\left(\nabla^2\psi_T - 2\lambda^2\psi_T\right)}{\partial t} = -\left[\mathbf{u}_m\nabla\right]\left(\nabla^2\psi_T - 2\lambda^2\psi_T\right) - \left[\mathbf{u}_T\nabla\right]\left(\nabla^2\psi_m + 2\lambda^2\psi_m\right) - \beta\frac{\partial\psi_T}{\partial x} = 0$$

$$\text{where } \mathbf{u}_m = \mathbf{U}_m + \mathbf{u}'_m = \begin{pmatrix} U_m \\ 0 \end{pmatrix} + \begin{pmatrix} {u'}_m \\ {v'}_m \end{pmatrix} \quad \text{and} \quad \mathbf{u}_T = \mathbf{U}_T + \mathbf{u}'_T = \begin{pmatrix} U_T \\ 0 \end{pmatrix} + \begin{pmatrix} {u'}_T \\ {v'}_T \end{pmatrix}$$

In this context, the prognostic variable for the second (baroclinic mode) equation is no longer $\nabla^2\psi_T$ but $\tilde{\zeta} = \nabla^2\psi_T - 2\lambda^2\psi_T$. The variable ψ_T (needed to obtain the perturbation velocities for the baroclinic mode and the beta effect is then obtain as a solution of a Helmholtz-type equation

$$\psi_T = \left[\nabla^2 - 2\lambda^2\right]^{-1}\tilde{\zeta}$$

[1] Note that with `rhs`, we are passing a function as argument, i.e. we are using the concept of function pointers

. In spectral space, this may be written as

$$\hat{\psi}_T = -\frac{1}{k^2+\lambda^2}\hat{\tilde{\zeta}}$$

. Proceed as follows with the implementation of the two-layer model:

- in analogy to the operator `poisson` in `utils.py`, define an operator `helmholtz` that takes an additional parameter `lambda` as argument and solves for the corresponding Helmholtz-type equation. Check the solution for accuracy with a harmonic field and $\lambda^2 = 1$.
- implement the solution in the dummy routine `two_layer` where $\zeta[0,:,:]$ corresponds to the barotropic mode and $\zeta[1,:,:]$ to the baroclinic mode, i.e. $\zeta[1,:,:] = \tilde{\zeta}$

b) Investigate the growth rate of a baroclinic instability!

c) Analyze the terms of the Lorenz energy cycle that play a role for the baroclinic instability!