Quasi-Geostrophic (QG) Models (Two-Layer and Barotropic)

Creator: Zhenyu You, Georgia Tech

Contact: zyou2998@gmail.com

Overview

This project provides two linearized Quasi-Geostrophic (QG) models for analyzing atmospheric instability:

1. **Two-Layer QG Model** – default levels at 250 and 750 mb, with heating added at 500 mb.
2. **Barotropic (One-Layer) QG Model**.

Both models calculate **normal modes** (eigenmodes of the dynamical system with exponential growth rates) and **optimal modes** (structures that maximize amplitude growth over a set optimization time).

Other features include spatially varying damping, local optimization, and (for the two-layer model) parameterized diabatic heating requiring heating sensitivity matrices (mat_*.nc).

Directory Structure

The models expect a specific directory structure. It is recommended to set up the following folders within your main project directory:

project_directory/

├── input_data/ # Input files: SST (for land-sea difference of damping magnitudes), background flow (GP_*), static stability (S2_*), heating map (mat_*.nc, if used).

├── output_data/ # All generated .nc files (matrices, modes).

├── pics/ # All generated plots (.pdf).

└── code/ # Python and shell scripts.

Paths to these directories (in_dir, out_dir, pic_dir) must be set at the top of model_parameters.py.

Workflow

Run the analysis from a single script:

1. **Configure Parameters:** Open the **model_parameters.py** file in the directory of the model you wish to run. All physical, numerical, and experimental parameters for that model are included in this file.
2. **Run Experiments:** Execute **./run_experiment.sh.** Multiple parameter sets (damping, heating, optimization times) can be specified. The script updates model_parameters.py, runs the analysis, and generates data/plots.

Key Scripts

The analysis for each model is broken down into a sequence of scripts called by run_experiment.sh.

- model_parameters.py: **Main configuration file.**
- toolbox.py: Numerical operators (e.g., discretization, Laplacian, derivatives).

Two-Layer Model Analysis Chain:

1. 00_damping_files.py: Generates separate damping fields for the upper and lower layers.
2. 00_effective_stability.py: Calculates the static stability and modifies it based on heating settings.
3. 01_LHS_Matrix.py: Constructs the 2x2 block LHS matrix operator.
4. 02_RHS_Matrix.py: Constructs the 2x2 block RHS matrix operator.
5. 03_normal_mode.py: Solves the eigenvalue problem for the normal modes.
6. 04_optimal_mode.py: Calculates the optimal modes and their evolution.

Barotropic Model Analysis Chain:

1. 00_damping_upper.py: Generates a single-layer damping field.
2. 01_LHS_Matrix.py: Constructs the simple LHS matrix.
3. 02_RHS_Matrix.py: Constructs the RHS matrix operator.
4. 03_normal_mode.py: Solves the eigenvalue problem for the normal modes.
5. 04_optimal_mode.py: Calculates the optimal modes and their evolution.

Plotting (Both Models):

- plot_normal_mode.py: Visualizes the spatial structure of the leading normal modes.
- plot_optimal_mode.py: Visualizes the time evolution of the leading optimal modes.