xomega Documentation

Release 0.1

Takaya Uchida

Jul 29, 2019

OVERVIEW

xomega is a Python package for solving the generalized omega equation which takes the form

$$N^2\nabla_{\rm h} w_{\rm b} + f_0^2 \frac{\partial^2 w_{\rm b}}{\partial z^2} = \beta \frac{\partial b}{\partial x} + \nabla_{\rm h} \cdot \mathbf{Q}(u, v, \theta, \Phi).$$

Given the right-hand side of the equation, the package inverts the vertical velocity fields at each time and depth in wavenumber space, i.e.

$$-\kappa^2 N^2 \hat{w}_{\rm b} + f_0^2 \frac{\partial^2 \hat{w}_{\rm b}}{\partial z^2} = ik\beta\hat{b} + (ik\hat{Q}_x + il\hat{Q}_y).$$

where $\kappa = (k, l)$ is the horizontal wavenumber vector. The right-hand side, neglecting the turbulent correlation terms is $\mathbf{Q} = \mathbf{Q}_{\rm tw} + \mathbf{Q}_{\rm da}$ where

$$\mathbf{Q}_{\rm tw} = -2\Big(\frac{\partial \mathbf{u}}{\partial x} \cdot \nabla b, \frac{\partial \mathbf{u}}{\partial y} \cdot \nabla b\Big)$$

$$\mathbf{Q}_{\rm da} = f\Big(\frac{\partial v}{\partial x}\frac{\partial u_{\rm a}}{\partial z} - \frac{\partial u}{\partial x}\frac{\partial v_{\rm a}}{\partial z}, \frac{\partial v}{\partial y}\frac{\partial u_{\rm a}}{\partial z} - \frac{\partial u}{\partial y}\frac{\partial v_{\rm a}}{\partial z}\Big).$$

where $\mathbf{u}_{\rm a}$ $(= \mathbf{u} - \mathbf{u}_{\rm g})$ is the ageotrophic velocity. Assuming the total flow to be in geostrophic balance $\mathbf{u} = \mathbf{u}_{\rm g} = \frac{\hat{z}}{f} \times \nabla_{\rm h}\Phi$, the generalized form reduces to the quasi-geostrophic Omega equation.

Note: xomega is at early stage of development and will keep improving in the future. The rigid-lid API should be quite stable, but minor utilities could change in the next version. If you find any bugs or would like to request any enhancements, please raise an issue on GitHub.

CHAPTER

ONE

CONTENTS

1.1 Current limitations

1.1.1 Uniform Vertical Grid

The algorithm used in xomega currently only supports input on uniform vertical grid. Calculation using data on non-uniform grid will give no error but the discretization error may be significant.

1.1.2 Rigid-lid Boundary Conditions

Currently, only rigid-lid boundary conditions are implemented.

1.2 Installation

1.2.1 The quickest way

xomega is compatible both with Python 2 and 3. The major dependencies are xarray and dask. The best way to install them is using Anaconda and pip.:

```
$ conda install xarray dask
$ pip install xomega .
```

1.2.2 Install xomega from GitHub repo

To get the latest version:

```
$ git clone https://github.com/roxyboy/xomega.git
$ cd xomega
$ python setup.py install .
```

Developers can track source code changes by:

```
$ git clone https://github.com/roxyboy/xomega.git
$ cd xomega
$ python setup.py build .
```

1.3 API

1.3.1 xomega

xomega.xomega.**w_rigid**(*N2*, *f0*, *beta*, *Frhs*, *kx*, *ky*, *dZ*, *dZ0=None*, *dZ1=None*, *zdim='Zl'*, *dim=None*, *coord=None*)

Inverts the Omega equation given by Giordani and Planton (2000) to get the balanced vertical velocity (w_b) for rigid-lid boundary conditions given the right-hand side fo the equation.

Parameters

N2 [float or xarray.DataArray] The buoyancy frequency.

f0 [float] Coriolis parameter.

beta [float] Meridional gradient of the Coriolis parameter for a beta-plane approximation.

Frhs [xarray.DataArray] The Fourier transform of the right-hand side of the Omega equation. The last two dimensions should be the meridional and zonal wavenumber.

kx [xarray.DataArray] Zonal wavenumber.

ky [xarray.DataArray] Meridional wavenumber.

dZ [float or xarray.DataArray] Vertical distance between grid. This should take constant value(s) for best numerical percision.

dZ0 [float or xarray.DataArray, optional] Top vertical distance between grids.

dZ1 [float or xarray.DataArray, optional] Bottom vertical distance between grids.

zdim [str, optional] Dimension name of the vertical axis of *Frhs*.

dim [list, optional] List of the xarray.DataArray output.

coord [dict, optional] Dictionary of the xarray.DataArray output.

Returns

wa [xarray.DataArray] The inverted ageostrophic vertical velocity.

PYTHON MODULE INDEX

x

INDEX

W

X