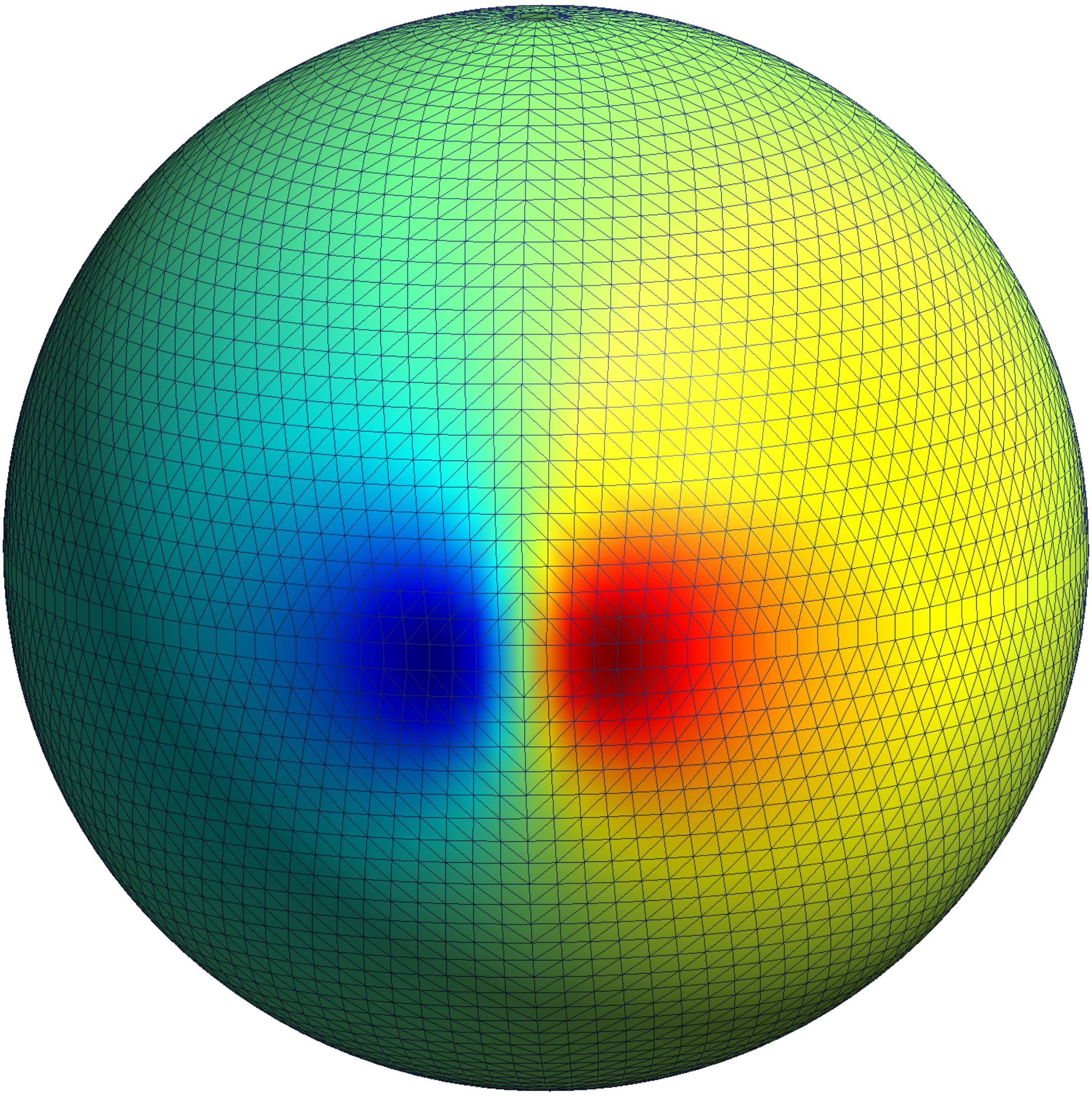
Elliptic Solver Dwarf

Hands-on exercises for the course
Advanced Numerical Methods



Introduction

We study in these exercises the three dimensional version of the elliptic solver dwarf. This code solves a potential flow over a Gaussian-shaped hill.

Configuration File

The following table gives a short explanation of the different parameters in the configuration file under the section "general":

name	description	default value
gridID	the letters at the beginning of this value determine the type of the mesh that is used for the simulation. Possible values are: "O" for octahedral mesh and "Slat" for longitude-latitude mesh. The number at the end of the value determines the resolution. This number gives the number of latitudes in one hemisphere. The default value "O32" for example uses an octahedral mesh with 32 latitudes in one hemisphere. This is approximately one quarter of the number of grid cells along the equator.	O32
nb_levels	number of levels in the vertical direction	51
dz	vertical resolution in meters	800
planet_radius	radius of the planet in meters	6.37122e+06
hill_radius	radius of the hill in meters	3.0e+06
hill_height	height of the hill in meters	4000
vstretch	0: no stretching (vertical resolution is constant), 1: with stretching (vertical resolution becomes finer towards the bottom. The parameter dz describes in this case the average vertical resolution)	1
vx0	wind speed in m/s of the ambient velocity field along the equator	20.0
nb_precon_iter	number of preconditioner iterations	3
eps0	iterative solution is stopped if the residuum is smaller than this tolerance eps0	1.0e-8
kord	order of the method	3
itmn	minimum number of iterations	1
itmx	maximum number of iterations	60

Exercises

1. Running the code

Open Terminal and go to your home folder with the command cd. Execute the following command to copy the folder necessary to run the code to your home folder:

```
cp -r /home/ectrain/trx/NM_TC2017/ellipticSolver $HOME/
```

Go into this folder with

```
cd ellipticSolver
```

You can now run the code with the command

```
./run out
```

This command runs the code and write the log message that are shown on screen also to the file out.

The setup of the simulation can be changed by editing the file `config.json` inside this folder `ellipticSolver`. If you have no favourite editor we recommend to open the file with

```
gedit config.json &
```

Please remember to save the file after making changes. You can plot the convergence stored in previously used output-files out1, out2, ... with

```
python plot_resid.py out1 out2 ...
```

Familiarise yourself with running the code, plotting the convergence and with the data that is shown on the screen while running the code.

2. Accuracy

Change the accuracy threshold "eps0" and see how it impacts the number of iterations required to converge to that threshold?

3. Hill height

Change the mountain height "hill_height" and see how it impacts convergence. What do you observe for a zero hill height?

4. Vertical resolution

How does the convergence change if you reduce the height of the atmosphere by reducing "dz"?

5. No preconditioner

Switch off the preconditioner by setting "nb_precond_iter" to 0. What do you observe?

6. Small planet

While keeping the preconditioner switched off and the vertical resolution at 800m, what do you observe if you reduce the radius of the planet down to ~50km (keep in mind that the mountain width needs to be adjusted accordingly)?

7. Optional for students with experience in Fortran

You can take a look at the source code of the dwarf in the following directory:

```
/home/ectrain/trx/NM_TC2017/dwarf-D-ellipticSolver-GCR/sources/
dwarf-D-ellipticSolver-GCR/src/prototype1
```