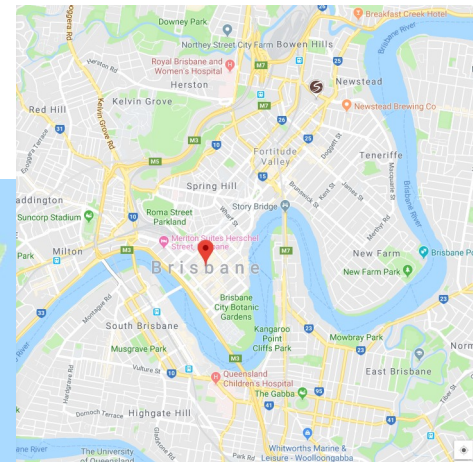

The University of Queensland
School of Earth and Environmental Sciences

**Geophysical Imaging with Finite Elements and
Python**

Lutz Gross
E-mail : l.gross@uq.edu.au



Where is Home?



Course Outline

- AIM: Hands-on introduction to
 - Python
 - Finite element method and application in geophysics
 - Concepts of inversion
- Content:

Week 1 (3/11): Introduction to python (incl. practical with Jupyter notebooks)

Week 2 (10/11): More on python programming (incl. practical with Jupyter notebooks)

Week 3 (17/11): Introduction to the Finite Element Method (FEM)

Week 4 (24/11): FEM modeling in geophysics using esys-escript in python

Week 5 (1/12): Parameter Fitting (incl. practical with Jupyter notebooks)

Week 6 (8/12): The Concept of Inversion

Week 7 (15/2): FEM-based Inversion in Geophysics

Week 8 (22/12): Advanced Inversion Techniques



We use Open Source software

- This is software with source code that anyone can inspect, modify, and enhance
 - But watch out for the license: Apache, GPL, etc
- Developments are carried by communities
 - Need help: <https://www.python.org/about/help/>
 - Get help and help others!
- All documentation is on-line:
 - Not happy? Fix it!

Software for the Course

- What software do I need for the course?
 - Python3
 - Arrays: Numpy
 - Visualization: Matplotlib
 - numerical algorithms: Scipy
 - FEM solver: esys-escript
- Jupyter notebook
 - Edit & run scripts via Web-based server
 - Integrating text & code

Beyond Jupyter Notebook

- Integrated Development Environment (IDE)
 - PyCharm
 - PyDev with Eclipse
 - Spyder IDE
 - Python IDLE
 - etc

Course Material on GitHub

The screenshot shows the GitHub interface for the repository 'AndreaCodd / eenotes'. The repository is public and has 2 watchers. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights. The repository has a 'main' branch with 1 branch and 0 tags. A 'Clone' dropdown menu is open, showing options for HTTPS, SSH, and GitHub CLI. The HTTPS URL is 'https://github.com/AndreaCodd/eenotes.git'. Below the clone options, there are links to 'Open with GitHub Desktop' and 'Download ZIP'. The repository content shows a file named 'README.md' with a commit history of 'first commit'.

AndreaCodd / eenotes Public

Unwatch 2

<> Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags

Go to file Add file Code

Clone

HTTPS SSH GitHub CLI

<https://github.com/AndreaCodd/eenotes.git>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

LutzGross resubmit

File	Commit
Notebooks	resubmit
README.md	first commit

README.md

eenotes

Software Access

A) Python distribution anaconda

- Install via <https://www.anaconda.com/>
- Includes many python modules
- Limited esys-escript support :-(

B) install a web-server

- Via a Docker virtual machine
- See notes distributed

C) use cloud server

- Visit <https://esys-escript.github.io/>
- Click on Jupyter logo

esys-escript

tool for implementing mathematical models in python using the
structures it is very easy to use and scripts can run on desktop or
cloud changes. Application areas for escript include earth mantle convec-
tion, media flow, reactive transport, plate subduction, erosion, and tsunamis.

esys-escript online in a Jupyter Notebook. Choose your own user name.



easy-to-use environment for implementing mathematical models

Click here


Create An Account



- Don't forget your password!
- Download your data at one point ...

Navigate to Course Material

With preloaded course material

 jupyter

QuitLogout









FilesRunningClusters

Select items to perform actions on them.

UploadNew↺

☐ 0 ▾

/ [eenotes](#) / [Notebooks](#) / [GeophysicalImaging](#) / [material](#) / [Notebooks](#)

	Name ▾	Last Modified	File size
 ..		seconds ago	
<input type="checkbox"/>  Data		4 hours ago	
<input type="checkbox"/>  00_Intro.ipynb		2 hours ago	25.4 kB
<input type="checkbox"/>  01_PythonBasics.ipynb		2 hours ago	68.3 kB
<input type="checkbox"/>  02_Arrays.ipynb		2 hours ago	209 kB
<input type="checkbox"/>  11_FEM.ipynb		2 hours ago	97.4 kB
<input type="checkbox"/>  21_Gravity.ipynb		2 hours ago	178 kB
<input type="checkbox"/>  22_Magnetic.ipynb		2 hours ago	109 kB

After login

The screenshot shows the JupyterLab interface after a successful login. The interface includes a top navigation bar with 'Files', 'Running', and 'Clusters' tabs. Below this is a file browser showing a directory structure. A red arrow points from the 'Your files and directories' box to the file browser. Another red arrow points from the 'Upload file' box to the 'Upload' button. A third red arrow points from the 'Leave Jupyter' box to the 'Logout' button. A fourth red arrow points from the 'Create new notebook, directory' box to the 'New' button. A fifth red arrow points from the 'Click to start Notebook in new tab' box to the 'GravityVariationWithLatitude_notebook.ipynb' file. A sixth red arrow points from the 'Notebook running in other browser tab' box to the 'Running' status of the 'JupyterBasics.ipynb' file.

Your files and directories

Upload file

Leave Jupyter

Create new notebook, directory

Click to start Notebook in new tab

Notebook running in other browser tab

jupyter

Files Running Clusters

Select items to perform actions on them.

0 / 01_GravityVariationWithLatitude

Name	Last Modified
..	seconds ago
Output	5 minutes ago
GravityVariationWithLatitude_notebook.ipynb	an hour ago
JupyterBasics.ipynb	Running 7 minutes ago
solution_GravityVariationWithLatitude.ipynb	43 minutes ago
Earth.png	a year ago
hammer_paper.pdf	a year ago

Jupyter notebook

- web application → use your favorite browser
 - contain live python code
 - includes narrative text
 - contains equations
 - includes visualizations
 - to create and share documents
- See <https://jupyter.org/>



A jupyter notebook

- is two things:
 - A document containing text and python code
 - Extension: ipynb
 - A session
 - to render the text
 - to run the code/sections of the code

notebook page

The screenshot shows a Jupyter Notebook interface with the following components and annotations:

- Header:** "jupyter 00_Intro (unsaved changes)" and a "Logout" button.
- Menu Bar:** File, Edit, View, Insert, Cell, Kernel, Help.
- Toolbar:** Includes icons for saving, adding cells, undo, redo, and a "Run" button. A red box labeled "Run python code + render cells → active cell only" points to the "Run" button.
- Section Header:** "Jupyter Basics".
- Text:** "This is some basic python code:".
- Code Cells:**
 - Cell 1: `In [1]: from math import sqrt`, `x=sqrt(2.)`, `print("x =",x)`. The output is `x = 1.4142135623730951`. A red box labeled "Stop Running python code" points to the "Run" button, and a red box labeled "Cell with text" points to this cell.
 - Cell 2: `In [2]: x**2`. The output is `Out[2]: 2.0000000000000004`. A red box labeled "active cell with code" points to this cell.
 - Cell 3: `In []: x**3`. A red box labeled "Cell with python code" points to this cell.
- Section Header:** "Matplotlib".
- Text:** "graphical output is shown in the notebook after run:".
- Code Cell:** `In []: import numpy as np`, `import matplotlib.pyplot as plt`.

Cells after 'run'

The screenshot shows the Jupyter interface with the following elements:

- Header:** "jupyter JupyterBasics Last Checkpoint: Last Wednesday at 1:56 PM (unsaved changes)"
- Menu Bar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help
- Toolbar:** Includes icons for saving, adding cells, and running. A dropdown menu is open, showing "Code" as the selected cell type.
- Cell 1:** Contains Python code:

```
In [1]: 1 from math import sqrt
2 x=sqrt(2.)
3 print("x =",x)

x = 1.4142135623730951
```
- Cell 2:** Contains Python code:

```
In [3]: 1 x**2

Out[3]: 2.0000000000000004
```
- Cell 3:** An empty code cell with the prompt "In []: 1".

Red arrows indicate the following actions:

- An arrow points from the "Code" dropdown menu to the text box: "Change cell type Code or text ('markdown')".
- An arrow points from the first code cell to the text box: "All cells treated as single python code".
- An arrow points from the third empty cell to the text box: "Cell code cell added".

All cells treated as
single python code

Cell code cell
added

Matplotlib output is integrated

Out[14]: 2.8284271247461907

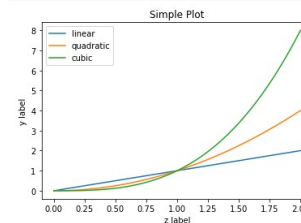
```
1 # Matplotlib
2
3 graphical output is shown in the notebook after run:
```

```
In [ ]: 1 import numpy as np
2 import matplotlib.pyplot as plt
3 # create 100 points from 0 to 2 (including 2)
4 z = np.linspace(0, 2, 100)
5 plt.plot(z, z, label='linear')
6 plt.plot(z, z**2, label='quadratic')
7 plt.plot(z, z**3, label='cubic')
8 plt.xlabel('z label')
9 plt.ylabel('y label')
10 plt.title("Simple Plot")
11 plt.legend()
12 plt.show()
```

Matplotlib

graphical output is shown in the notebook after run:

```
In [15]: 1 import numpy as np
2 import matplotlib.pyplot as plt
3 # create 100 points from 0 to 2 (including 2)
4 z = np.linspace(0, 2, 100)
5 plt.plot(z, z, label='linear')
6 plt.plot(z, z**2, label='quadratic')
7 plt.plot(z, z**3, label='cubic')
8 plt.xlabel('z label')
9 plt.ylabel('y label')
10 plt.title("Simple Plot")
11 plt.legend()
12 plt.show()
```



File Menu



Example report

Jupyter Basics

August 26, 2019

1 Jupyter Basics

This is some basic python code:

```
In [12]: from math import sqrt
x=sqrt(2.)
print("x =",x)

x = 1.4142135623730951
```

```
In [13]: x**2
```

```
Out[13]: 2.0000000000000004
```

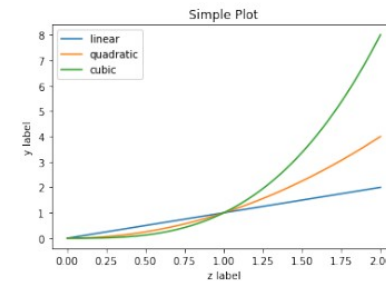
```
In [14]: x**3
```

```
Out[14]: 2.8284271247461907
```

2 Matplotlib

graphical output is shown in the notebook after run:

```
In [15]: import numpy as np
import matplotlib.pyplot as plt
# create 100 points from 0 to 2 (including 2)
z = np.linspace(0, 2, 100)
plt.plot(z, z, label='linear')
plt.plot(z, z**2, label='quadratic')
plt.plot(z, z**3, label='cubic')
plt.xlabel('z label')
plt.ylabel('y label')
plt.title('Simple Plot')
plt.legend()
plt.show()
```



And more

- ‘Edit’ → copy & past & split & merge
- ‘Insert’ → add new cell
- ‘Run’ → run all cells, run cells above/below, ...
- For markdown text see for instance:
 - <https://www.datacamp.com/community/tutorials/markdown-in-jupyter-notebook>