Scientific computation using python

Ash Suresh

September 11, 2024

python

- Relatively new language, created by Guido von Rossum in 1992.
- The name doesn't imply to the snake, but rather is inspired from a British comedy troupe, Monty Python.

Why python?

- Simple, tailored for teaching.
- Simple to learn.
- Lower dev time.
- Contains many useful features of many languages.

Why python?

- Simple, tailored for teaching.
- Simple to learn.
- Lower dev time.
- Contains many useful features of many languages.
- Completely general programming language.

Why python?

- Simple, tailored for teaching.
- Simple to learn.
- Lower dev time.
- Contains many useful features of many languages.
- Completely general programming language.
- It was written to be "easy to learn"

Why python?

- Simple, tailored for teaching.
- Simple to learn.
- Lower dev time.
- Contains many useful features of many languages.
- Completely general programming language.
- It was written to be "easy to learn"
- Hence it is built to be readable, not fast.

Why python?

- Simple, tailored for teaching.
- Simple to learn.
- Lower dev time.
- Contains many useful features of many languages.
- Completely general programming language.
- It was written to be "easy to learn"
- Hence it is built to be readable, not fast.
- We will learn python "on the job", so to speak.

Code Writing Strategies: Simplicity, Readability, Repeatability.

Code Writing Strategies: Simplicity, Readability, Repeatability.

- Dumb code that exists is better than non-existant clever code.

Code Writing Strategies: Simplicity, Readability, Repeatability.

- Dumb code that exists is better than non-existant clever code.
- Premature optimisation is the root of all EVIL!!!

Code Writing Strategies: Simplicity, Readability, Repeatability.

- Dumb code that exists is better than non-existant clever code.
- Premature optimisation is the root of all EVIL!!!
- Test as you write, write as you test.

Code Writing Strategies: Simplicity, Readability, Repeatability.

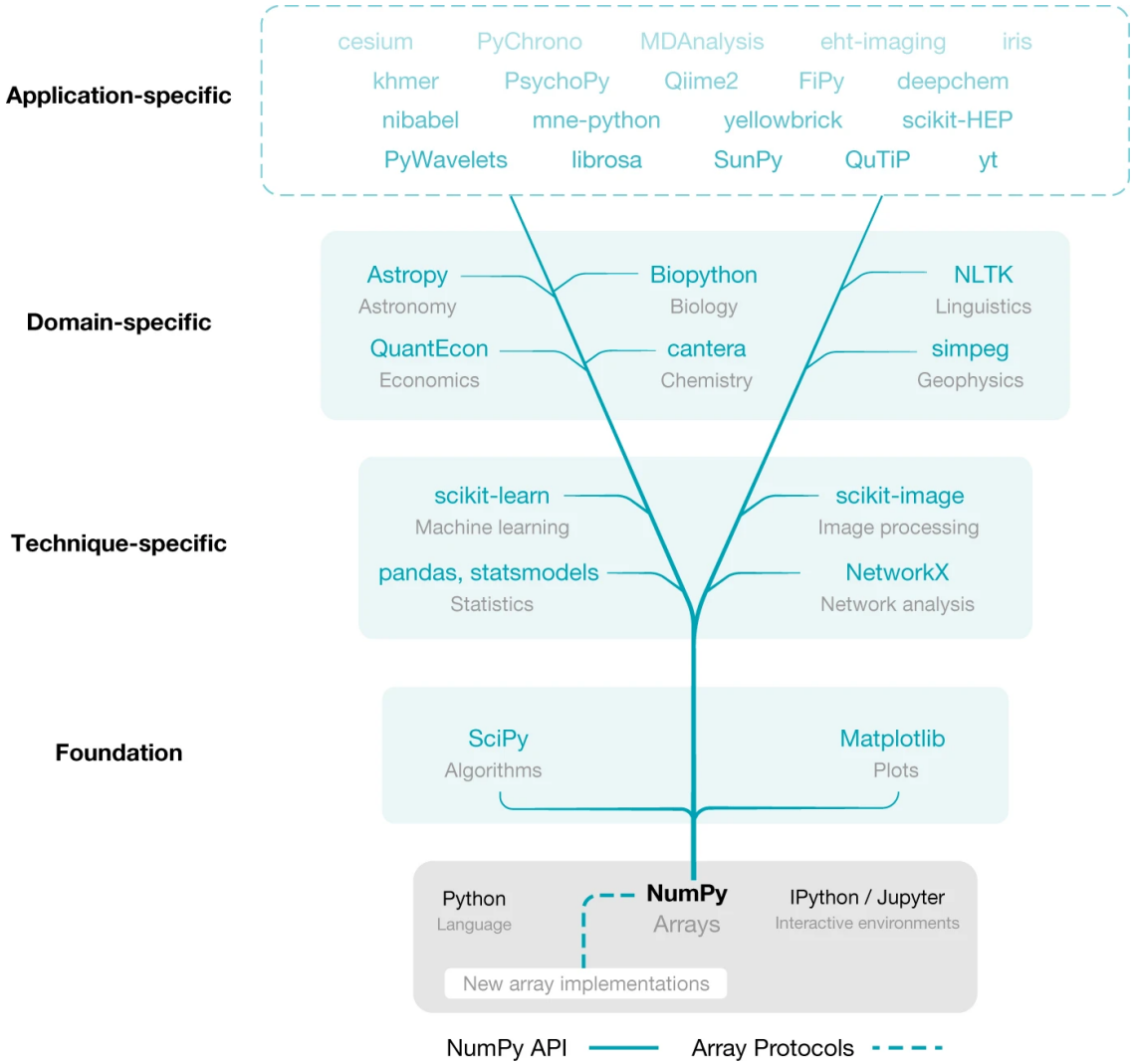
- Dumb code that exists is better than non-existant clever code.
- Premature optimisation is the root of all EVIL!!!
- Test as you write, write as you test.
- Divide and Conquer. There are no difficult problems, there are just too many simple problems.

Why python?

- High level language, low development time, easier debugging and maintenance.
- Free, large community, well documented.
- A large collection of libraries.

Why python?

Nature volume 585, pages 357–362 (2020)



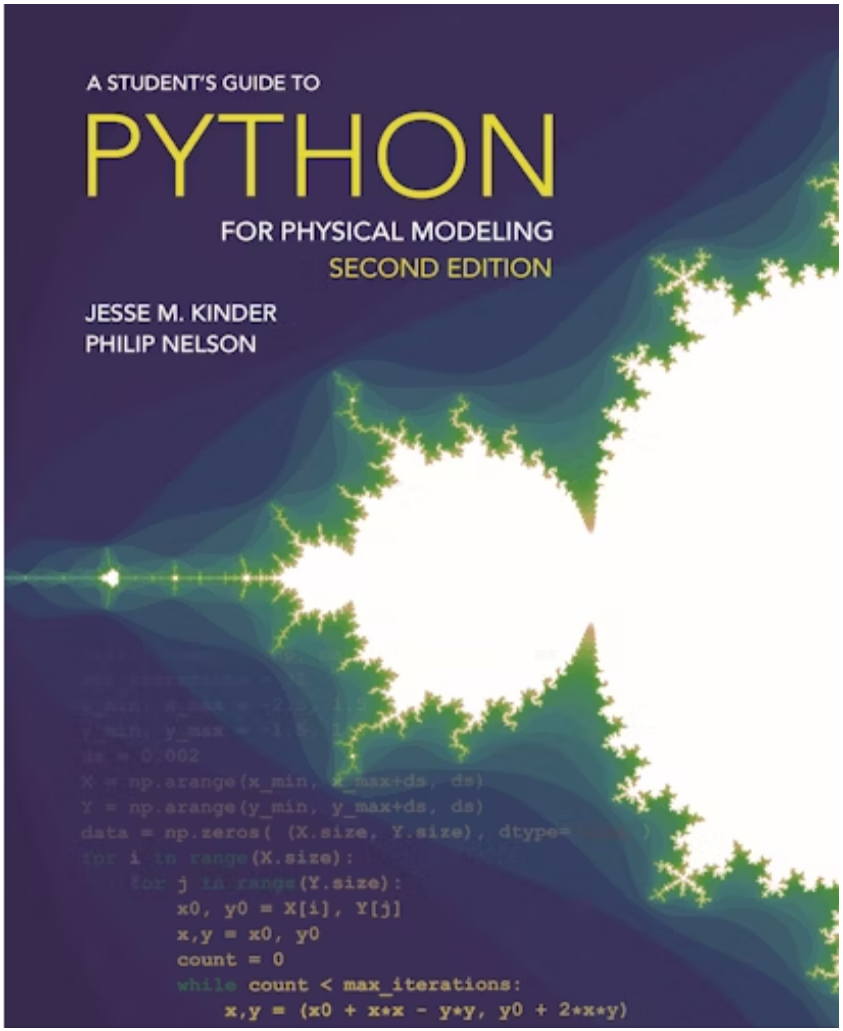
Getting started.

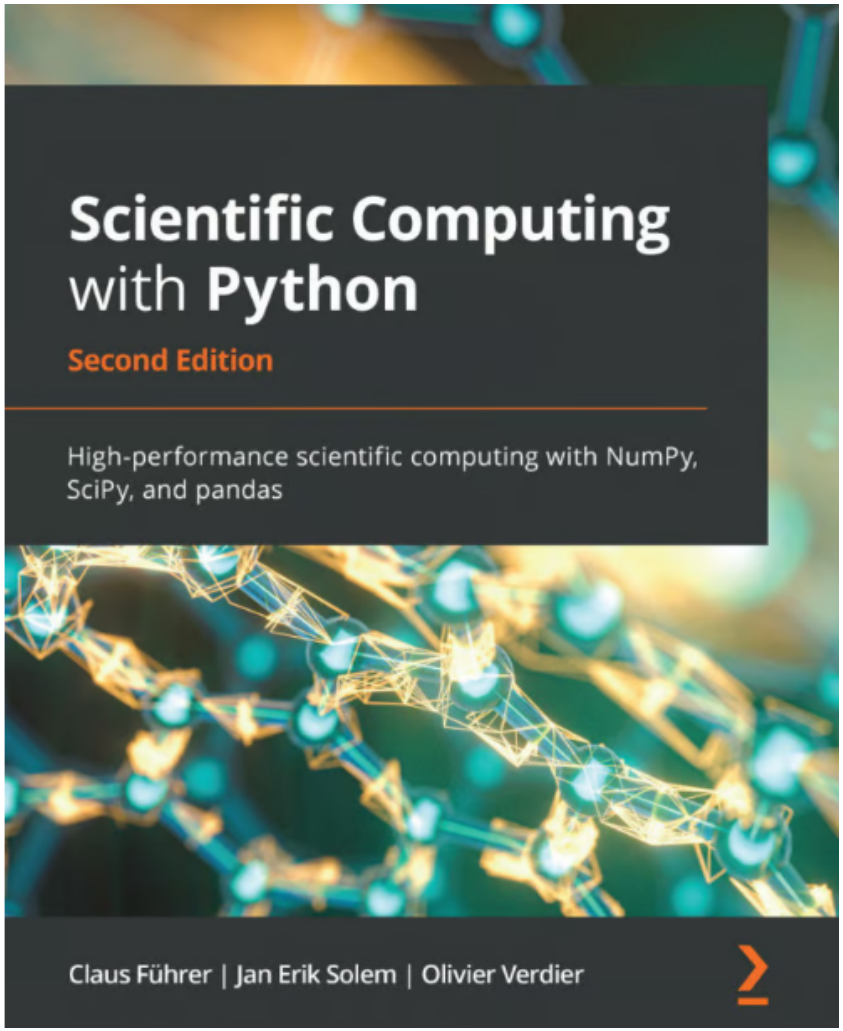
- We will be running codes interactively using jupyter notebooks.
- Those using unix systems can install conda and then install and run python.
- If you can't install on your computer, don't worry, python can be run on cloud.

Topics that will be covered.

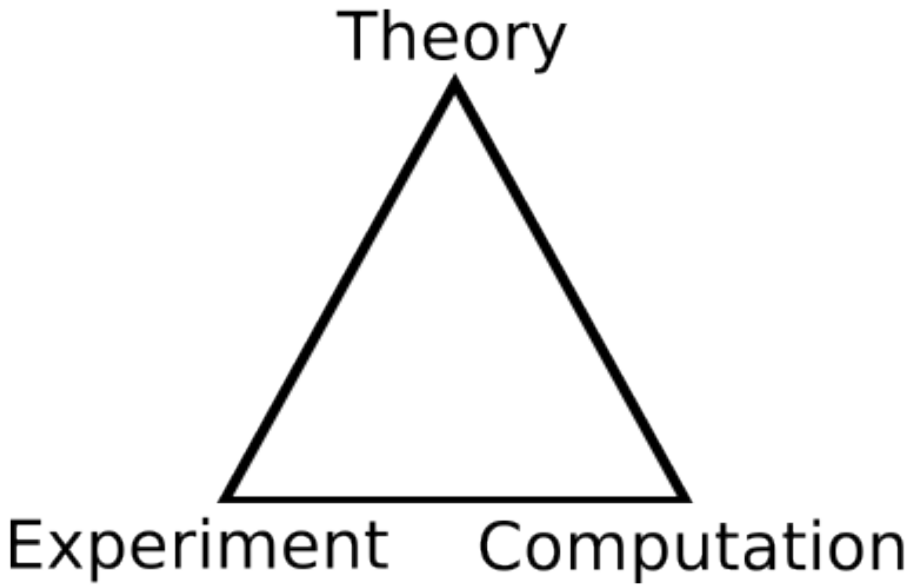
1. Basic python with application for physics or science based examples.
2. numpy, matplotlib, scipy, sympy, pytorch, pandas, seaborn

References


A STUDENT'S GUIDE TO
PYTHON
FOR PHYSICAL MODELING
SECOND EDITION
JESSE M. KINDER
PHILIP NELSON
x0, y0 = X[i], Y[j]
x,y = x0, y0
count = 0
while count < max_iterations:
x,y = (x0 + x*x - y*y, y0 + 2*x*y)


Scientific Computing
with Python
Second Edition
High-performance scientific computing with NumPy, SciPy, and pandas
Claus Führer | Jan Erik Solem | Olivier Verdier

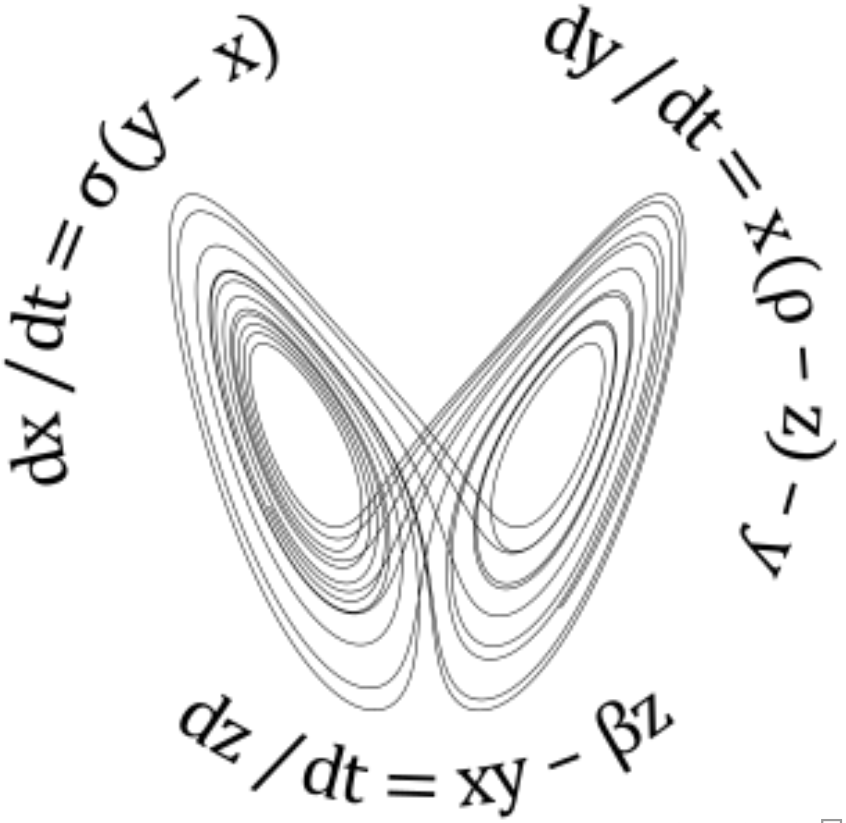
Computing in science



1. Theory: Formulation and analysis of idealized models and hypotheses to describe real systems.
2. Experiment: Quantitative measurement of real phenomena.
3. Computation: Computer experiments on models.
    - New third branch of science.

Computational science

1. Data analysis: manage/collect large amounts of data from experiments, fit theoretical and statistical models, graphical representation and other visualizations.
2. Numerical analysis: a lot of equations can't be solved analytically, one must resort to numerical solutions. Eg. Lorenz attractor,



Computational science

1. Data analysis: manage/collect large amounts of data from experiments, fit theoretical and statistical models, graphical representation and other visualizations.
2. Numerical analysis: a lot of equations can't be solved analytically, one must resort to numerical solutions. or even simple equation like this

$$tan(\theta) = \theta$$

3. Computer simulations/experiments: Numerically simulate physical systems using computers. Eg. Three body problem, Schrodinger equation of multi-electron systems etc.