Numeric Computation with Numpy and SciPy

1. Numeric Computing in Python
   - NumPy and SciPy in Python
   - the computational ecosystem
2. Scientific Computing
   - linear algebra
   - differential equations

MCS 320 Lecture 40
Introduction to Symbolic Computation
Jan Verschelde, 26 July 2024

Numeric Computation with Numpy and SciPy

1. Numeric Computing in Python
   - NumPy and SciPy in Python
   - the computational ecosystem

2. Scientific Computing
   - linear algebra
   - differential equations

NumPy and SciPy in Python

NumPy defines vectors and matrices,
is a foundational project for the
Python scientific computing stack.

SciPy provides fundamental algorithms
for scientific computing in Python.

`https://numfocus.org`

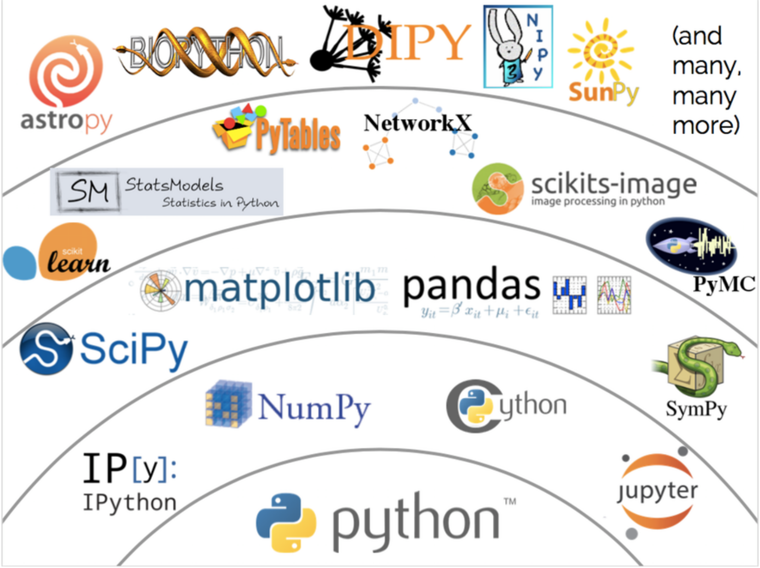
Numeric Computation with Numpy and SciPy

1. Numeric Computing in Python
   - NumPy and SciPy in Python
   - the computational ecosystem

2. Scientific Computing
   - linear algebra
   - differential equations

The Computational Ecosystem

picture from the slides of Jake VanderPlas, 2015



Numeric Computation with Numpy and SciPy

1. Numeric Computing in Python
   - NumPy and SciPy in Python
   - the computational ecosystem

2. Scientific Computing
   - linear algebra
   - differential equations

Linear Algebra

- NumPy provides vectorized functions, functions that take an entire vector on input and return a vector on output.

  Vectorized code speeds up Python code significantly.

- The backslash operator $\backslash$ as in $\mathbf{x} = A \backslash \mathbf{b}$ solves the linear system $A\mathbf{x} = \mathbf{b}$.

  The solvers are implemented in optimized software libraries.

- Arrays are data structures, matrices are arrays with defined linear algebra methods.

Numeric Computation with Numpy and SciPy

1. Numeric Computing in Python
   - NumPy and SciPy in Python
   - the computational ecosystem

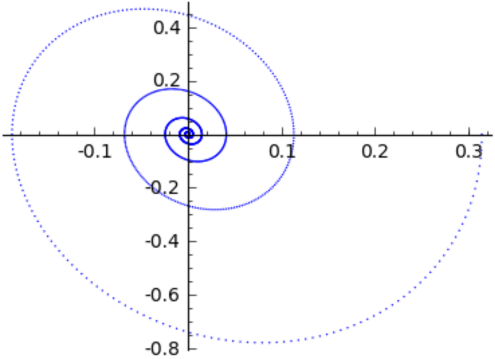
2. Scientific Computing
   - linear algebra
   - differential equations

Differential Equations

a model of a pendulum

$$\frac{d^2}{d\,t^2}\theta(t) = -\frac{d}{d\,t}\theta(t) - g\,\sin(\theta(t)), \quad \theta(0) = \pi/10, \quad \theta'(0) = 0$$

Plotting both displacement θ and velocity θ' gives a phase portrait:


0.4
0.2
-0.1
0.1
0.2
0.3
-0.2
-0.4
-0.6
-0.8