

# Introductory Scientific Computing with Python

Introduction to Python

FOSSEE

Department of Aerospace Engineering  
IIT Bombay

Mumbai, India

# Acknowledgement

**FOSSEE group** ([fossee.in](http://fossee.in))  
based at  
**IIT Bombay**  
and funded by  
The National Mission on Education  
through ICT,  
**Ministry of HRD, India**

# Why Python?

## For Scientific Computing?

# Need: Toolkit for diversity

- Numeric and Symbolic
- Exploration and Visualization
- High performance
- Parallel computing
- User interfaces, Web
- Other tasks

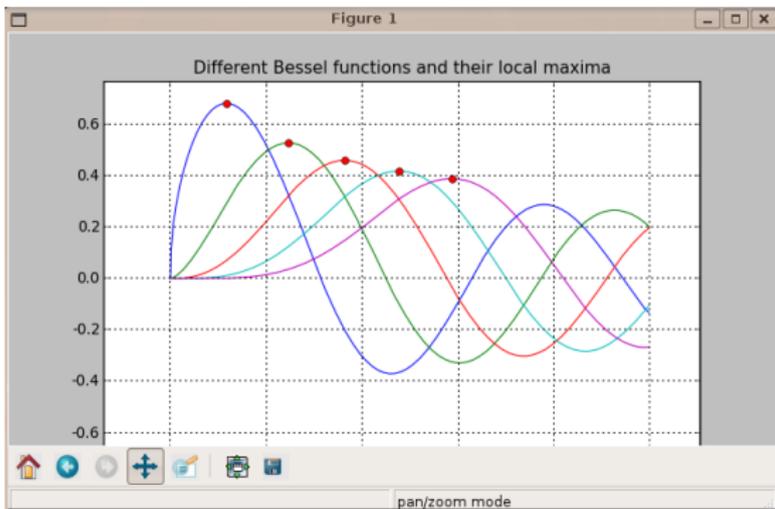
# Why Python?

- Easy to read and learn
- Powerful interactive interpreter
- Scalable, general purpose
- High-level, modular
- Procedural, OO, functional

# Why Python?

- Extensive libraries
- Rapid application development
- Interface to C++, C and FORTRAN
- Cross-platform
- Open Source

# Requirements: numeric computation



```
Welcome to pylab, a matplotlib-based Python environment.  
For more information, type 'help(pylab)'.
```

```
In [1]: from scipy import *  
In [2]: from numpy import *  
In [3]: from pylab import *  
In [4]: x = arange(0,10,0.01)  
In [5]: for k in arange(0.5,5.5):  
...:     y = special.jv(k,x)  
...:     p = plot(x,y)  
...:     f = lambda x: -special.jv(k,x)  
...:     x_max = optimize.fminbound(f,0,6)  
...:     p = plot([x_max], [special.jv(k,x_max)],'ro')  
...:
```

# SciPy

- Linear algebra
- Numerical integration
- Fourier transforms
- Signal processing
- Special functions
- Statistics
- Optimization
- Image processing
- ODE solvers
  
- Uses LAPACK, QUADPACK, ODEPACK, FFTPACK etc. from netlib

# Requirement: Exploration/Visualization

The image shows a screenshot of a computer interface with two windows. The left window is a terminal titled "Terminal" showing the execution of a Python script using IPython. The right window is titled "Mayavi Scene 1" and displays a 3D visualization of a scalar field. The visualization consists of a central cyan sphere surrounded by several curved, ribbon-like surfaces in shades of green and yellow. The Mayavi interface includes a pipeline view on the left, a toolbar at the top, and a control panel on the right.

```
Terminal  
resting ~ $ ipython -wthread -nobanner  
  
In [1]: from enthought.mayavi import mlab  
  
In [2]: from numpy import ogrid, sin  
  
In [3]: x, y, z = ogrid[-10:10:100j, -10:10:100  
  
In [4]: ctr = mlab.contour3d(0.5*x**2 + y**2 +
```

Mayavi pip

Pipeline

- Mayavi Scene 1
  - ScalarField
    - Colors and legends
      - IsoSurface

Scalar LUT Vector

LUT (Look Up Tab

Lut mode

Number of colors

Reverse lut

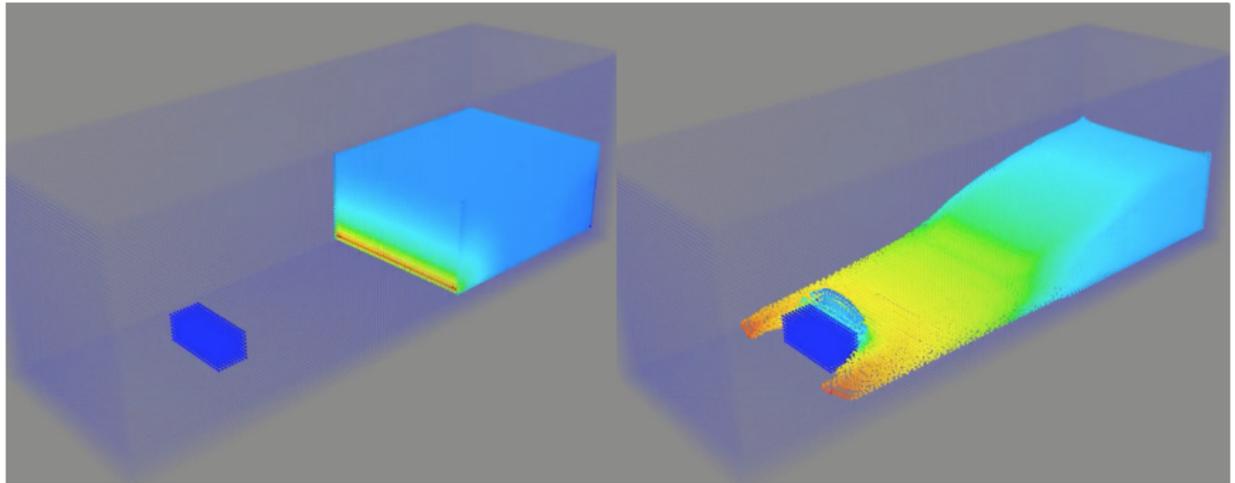
Show scalar bar:

# 3D visualization with Mayavi2

The screenshot displays the Mayavi2 application window. The top menu bar includes Python, File, Visualize, View, Tools, Window, and Help. The main window is titled "Mayavi2" and contains a "TVTK Scene 1" view. The left sidebar shows a hierarchical tree of objects: TVTK Scene 1, VTK XML file (fire Ug.vtu), Modules, Outline, Contour, PolyDataNormals, SetActiveAttribute, and Surface. The central 3D view shows a wireframe box containing three colored surfaces representing a fire simulation. The bottom-left panel is the "Mayavi object editor" with tabs for Contours, Actor, and Texturing. The "Contours" tab is active, showing settings for "Enable Contours" (checked), "Filled contours" (checked), "Auto contours" (checked), "Number of contours" (10), "Minimum contour" (307.84), "Maximum contour" (307.84), and "Auto update range" (checked). The bottom-right panel is a "Python" console showing the following text:

```
Python
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Startup script executed: /Users/prabhu/.python
>>>
```

# Requirement: HPC, parallel computing



# Requirement: UI

wxPython: (A Demonstration)

ListCtrl\_edit

- ▶ Frames and Dialogs
- ▶ Common Dialogs
- ▶ More Dialogs
- ▶ Core Windows/Controls
- ▼ Custom Controls
  - AnalogClockWindow
  - ColourSelect
  - Editor
  - GenericButtons
  - GenericDirCtrl
  - LEDNumberCtrl
  - MultiSash
  - PopupControl
  - PyColourChooser
  - TreeListCtrl
- ▶ More Windows/Controls
- ▶ Window Layout
- ▶ Process and Events
- ▶ Clipboard and DnD
- ▶ Using Images
- ▼ Miscellaneous
  - ColourDB
  - DrawXXXList
  - FileHistory
  - FontEnumerator
  - Joystick
  - OGL
- ▶ PrintFramework
- ▶ ShapedWindow
- ▶ Sound
- ▶ Unicode
- ▶ Samples using an external lib
  - Check out the samples dir too

OGL Overview Demo Code Demo

OGL is now a pure Python lib!

```
graph TD; Composite[Composite] --> Circle((Circle)); Composite --> Rectangle[Rectangle]; Composite --> DividedShape1[DividedShape]; RoundedRect[Rounded Rect] --> Polygon1{{Polygon}}; RoundedRect --> DividedShape2[DividedShape]; DrawnShape[DrawnShape] --> DrawText[DrawText]; Division[Division] --> Circle2((Circle)); Division --> Polygon2{{Polygon}};
```

Circle

Rectangle

DrawnShape

DrawText

Rounded Rect

DividedShape

This is Region number two.

Region 3 with embedded line breaks

Composite

Division

Polygon

OnAppActivate: False

OnActivate: True

OnAppActivate: True

OnActivate: False

OnAppActivate: True

OnActivate: False

OnAppActivate: True

OnAppActivate: True

# Super-simple UIs

```
from traits.api import *  
class Person(HasTraits):  
    name = Str('name')  
    age = Range(0.0, 200.0)  
    sex = Enum('male', 'female')  
  
p = Person(name='Ram')  
p.configure_traits()
```

The screenshot shows a wxPython GUI dialog box for configuring a Person object. The dialog has a light gray background and contains the following elements:

- Age:** A range slider with green arrowheads. The left value is 0.000 and the right value is 10.000. A blue circle is positioned on the slider. To the right of the slider is a text input field containing 0.000.
- Name:** A text input field containing the text "Ram".
- Sex:** A dropdown menu with "male" selected.
- Buttons:** "OK" and "Cancel" buttons are located at the bottom of the dialog.

The Web framework for perfectionists with deadlines.

Django makes it easier to build better Web apps more quickly and with less code.

## Meet Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.

Developed four years ago by a fast-moving online-news operation, Django was designed to handle two challenges: the intensive deadlines of a newsroom and the stringent requirements of the experienced Web developers who wrote it. It lets you build high-performing, elegant Web applications quickly.

Django focuses on automating as much as possible and adhering to the [DRY principle](#).

Dive in by [reading the overview](#) →

When you're ready to code, read the [installation guide](#) and [tutorial](#).

## The Django framework

### Object-relational mapper

Define your [data models](#) entirely in Python. You get a rich, [dynamic database-access API](#) for free — but you can still write SQL if needed.

### Automatic admin interface

Save yourself the tedious work of creating interfaces for people to add and update content. Django does that automatically, and it's production-ready.

## Download

Latest release: **1.1**

Open source, [BSD license](#)

## Documentation

[Installation guide](#)

[Tutorial](#)

[Using Django](#)

[Reference](#)

[More...](#)

## Sites that use Django

### lawrence.com

An internationally renowned local-entertainment site with events, stories, bands, drink specials and more.

### washingtonpost.com

The Washington Post's growing selection of innovative Web database applications.

### EveryBlock

A news feed for your block.

### LJWorld.com

## Weblog

### Django 1.1 released

by James Bennett on Jul. 29, 2009

Tonight the Django team is pleased to announce the release of Django 1.1, the latest and greatest stable version of the Django framework.

[Read more](#)

### Security updates released

by James Bennett on Jul. 28, 2009

In accordance with [our security policy](#), today the Django project is issuing a set of releases to remedy a vulnerability reported to us. This announcement contains a description of the vulnerability, a description of the changes made to fix it, and pointers to the patches for each supported version of Django.

Also covered here is an unrelated issue which, though security-related and resulting in changes to future Django releases, is not being treated as a vulnerability in Django itself.

[Read more](#)

### Django 1.1 release candidate available

by James Bennett on Jul. 21, 2009

Tonight the Django team has issued Django 1.1

# Easy to read and still compact?

```
def qsort(L):  
    """Quick sort for given sequence, 'L'."""  
    if not L: return L # exit recursion if input is empty  
    pivot, rest = L[0], L[1:]  
    less_than = [ lt for lt in rest if lt < pivot ]  
    greater_eq = [ ge for ge in rest if ge >= pivot ]  
    return qsort(less_than) + [pivot] + qsort(greater_eq)
```

Python users?



CIVILIZATION IV

NOKIA



Google™



journyx



WALT DISNEY  
FEATURE  
ANIMATION



INDUSTRIAL  
LIGHT & MAGIC  
A LUCASFILM COMPANY

You Tube



thawte

Diverse needs, one language

Python!



[www.python.org](http://www.python.org)