



# Python Tools

Prof. Venki Muthukumar, Ph.D.

# Anaconda

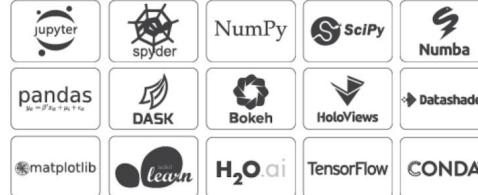


UAV Workshop

The screenshot shows the Anaconda Python/R Distribution website. The header includes a logo, navigation links for Products, Why Anaconda?, Solutions, Resources, Company, Contact Us, Download, and a search icon. The main section features a green background with a geometric pattern. The title "Anaconda Distribution" is prominently displayed, followed by the subtitle "The World's Most Popular Python/R Data Science Platform". A "Download" button is visible. The URL in the browser bar is [anaconda.com/distribution/](https://anaconda.com/distribution/).

The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with Conda
- Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
- Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
- Visualize results with Matplotlib, Bokeh, Databricks, and Holoviews



Windows | macOS | Linux

The screenshot shows the Anaconda 2019.10 for macOS Installer window. It features a green header with the Anaconda logo and the text "Anaconda 2019.10 for macOS Installer". Below the header is a cookie consent banner with a shield icon, the text "This website uses cookies to ensure you get the best experience on our website.", a "Privacy Policy" link, and an "ACCEPT" button. The main body of the window is mostly blank, showing a light gray background.

# Other Distributions & IDE



- Python Distributions
  - Active Python, Anaconda, Enthought Canopy, WinPython, python(x,y), etc.
- IDEs
  - NINJA, Spyder, PyCharm, eclipse, Django, pyscripter, BeeWare, Atom, Sublime, VisualCode,

# Anaconda Navigator



UAV Workshop

ANACONDA NAVIGATOR

Signed in as Teams810 | Sign out | Refresh

Applications on base (root) | Channels

Home | Environments | Learning | Community | Documentation | Developer Blog | Twitter | YouTube | GitHub

**JupyterLab** 1.1.4 | **Jupyter Notebook** 6.0.1 | **Spyder** 4.0.0 | **Glueviz** 0.15.2 | **Orange 3** 3.23.1

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

**RStudio** 1.1.456 | **VS Code** 1.41.1

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

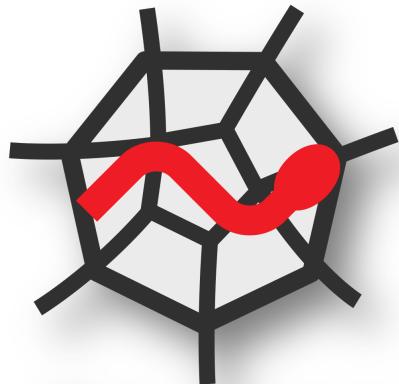
Streamlined code editor with support for development operations like debugging, task running and version control.

**Launch** | **Install** | **Launch** | **Install** | **Install**

**Documentation** | **Developer Blog**

**↑**

# SPYDER IDE



## SPYDER

The screenshot shows the Spyder IDE interface. At the top, there's a menu bar with "File", "Edit", "Cell", "Kernel", "Help", and a gear icon for settings. Below the menu is a toolbar with icons for file operations like Open, Save, and Run. The main area is divided into two panes: a code editor on the left containing Python code, and a terminal or output pane on the right showing command-line interactions. The title bar of the window says "Spyder - spyder".

spyder  
3.2.3

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

[Launch](#)



# SPYDER IDE



## What is Spyder?

Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language

# SPYDER IDE



UAV Workshop

Spyder (Python 3.7)

/Users/victor/spyder-py3/temp.py

temp.py

```
538
539     @contextmanager
540     def block_update_properties(self):
541         self._update_properties = False
542         yield
543         self._update_properties = True
544
545     def _update_coordinates(self):
546         """Insert the cursor position into the correct position in the
547         tuple of indices and update the cursor coordinates.
548
549         coords = list(self.dims.indices)
550         for d, p in zip(self.dims.displayed, self.position):
551             coords[d] = p
552         self.coordinates = tuple(coords)
553         self._value = self.get_value()
554         self.status = self.get_message()
555
556     def get_message(self):
557         """Generate a status message based on the coordinates and value
558
559         Returns
560
561         msg : string
562             String containing a message that can be used as a status update.
563
564         full_scale = np.multiply(self.scale, self._scale_view)
565         full_translate = np.add(self.translate, self._translate_view)
566
567         if any(
568             isinstance(n, slice) for n in self.coordinates
569         ): # Fixes bug where any element of self.coordinates is updated to a non-nl
570             self.coordinates = (0,) * len(self.coordinates)
571         full_coord = np.round(
572             np.multiply(self.coordinates, full_scale) + full_translate
573         ).astype(int)
574
575         msg = f'{self.name} {full_coord}'
576
577         value = self._value
578
579         if value is not None and not np.all(value == (None, None)):
580             msg += ': '
581             if type(value) == tuple:
582                 msg += status_format(value[0])
583                 if value[1] is not None:
```

Usage

Here you can get help of any object by pressing Cmd+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our [tutorial](#)

Variable explorer Help Plots Files

Console 1/A

Python 3.7.0 (default, Jun 28 2018, 07:39:16)  
Type "copyright", "credits" or "license" for more information.  
IPython 7.11.1 -- An enhanced Interactive Python.

In [1]:

IPython console History

conda base (Python 3.7.3), Line 556, Col 27 ASCII LF RW Mem 53%

# Spyder plugins



UAV Workshop

spydernotebook    GuitarModeling

Edit View Run Kernel Code Python 3 (Spyder)

[4]:

```
import numpy as np
from matplotlib import pyplot as plt

ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]

plt.plot(x, ys, 'r')
plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)

plt.title("Sample Visualization")
plt.show()
```

Sample Visualization

[6]:

```
#-----Parameters of the Simulation-----
n = 1
L = 0.6477 # String's length (m
d = 0.000309 # String's density E4 (kg/m
#d = 0.00578 # String's density E2 (kg/m
F = 56.34
#F = 65.9 # Tension (N)
c = np.sqrt(F/d) # Speed of waves in the string
frequency = (n*c)/(2*L)
Cn = 0.00005
t_0 = 0 #Initial time
deltaT = 0.0001
omega=frequency)
```

Variable explorer Help Plots Files

In [7]: `runfile('/Users/juanitagomez/.spyder-py3/temp.py', wdir='/Users/juanitagomez/.spyder-py3')`

In [8]: `runfile('/Users/juanitagomez/.spyder-py3/temp.py', wdir='/Users/juanitagomez/.spyder-py3')`

In [9]: `runfile('/Users/juanitagomez/.spyder-py3/temp.py', wdir='/Users/juanitagomez/.spyder-py3')`

In [9]: `print(d)`

Out[9]: `0.000309`

In [10]: `print(d)`

Out[10]: `array([199.3743578, 199.19178079, 200.49207342, 199.75735424,`

[...]

LSP Python: ready conda: spyder-dev-4 / Python 3.9.5, Line 1, Col 1 UTF-8 LF RW Mem 65%

# Installing Spyder Plugins



- Try
  - conda install spyder-notebook -c spyder-ide
  - conda install spyder-terminal -c spyder-ide
- else try
  - conda install spyder-notebook -c conda-forge
  - conda install -c conda-forge spyder-terminal

# The Notebook



A web-based interactive computing interface and platform that **combines code, equations, text and visualisations.**



<http://www.jupyter.org>

In a nutshell: an “interactive shell opened within the browser”  
10

Also called:

“Jupyter Notebook” or “IPython Notebook”

**No excuses possible when it comes to  
describe all steps in an analysis!**

# Anaconda - Ipython



A screenshot of the Anaconda Navigator application interface. The left sidebar includes links for Anaconda Navigator, File, Help, Home, Environments, Learning, Community, Documentation, and Developer Blog, along with social media icons for Twitter, YouTube, and GitHub. The main area shows a grid of applications:

Application	Version	Description	Action
Glueviz	0.15.2	Multidimensional data visualization across files. Explore relationships within and among related datasets.	Install
JupyterLab	1.2.5	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	Install
jupyter	6.0.2	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	Install
Orange 3	3.23.1	Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.	Install
Qt Console	4.6.0	PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	Install
RStudio	1.1.456	A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.	Install

# Jupyter Notebook



In a browser

The screenshot shows a web browser window for 'localhost:8888/tree'. At the top, there are tabs for 'Files', 'Running', and 'Clusters'. Below the tabs, a message says 'Select items to perform actions on them.' A list of notebooks is displayed:

- PresentationNotebooks
- cernbox
- HowTo\_ROOT-Notebooks.ipynb
- HowTo\_ROOT-Notebooks\_Long.ipynb
- My First Notebook.ipynb
- Untitled.ipynb

A callout box labeled 'A Choice of Kernels' points to the list of notebooks. To the right, a sidebar shows a dropdown menu with options: Upload, New, Text File, Folder, Terminal, Notebooks, Python 2, Python 3, and ROOT Prompt.

*Kernels* are processes that run interactive code in a particular programming language and return output to the user. Kernels also respond to tab completion and introspection requests.



A screenshot of a Jupyter Notebook interface. The browser title bar shows "localhost:8888/tree". The notebook toolbar includes File, Edit, View, Insert, Cell, Kernel, Help, and a dropdown for "Code" (Python 2). A red circle highlights the "Python 2" option. Below the toolbar, there's a toolbar with various icons for file operations like new, open, save, and cell controls. The main content area displays a large bold heading: "Welcome to the Notebook Technology". Below it, a text cell contains the instruction: "This is a markdown cell. You can add LaTex code:  $\sum_{n=-\infty}^{\infty} |x(n)|^2$ ".

**Text and  
Formulas**

Welcome to the Notebook Technology

This is a markdown cell. You can add LaTex code:

$$\sum_{n=-\infty}^{\infty} |x(n)|^2$$

```
In [1]: def thisFunction():
    return 42
```





localhost:8888/tree

File Edit View Insert Cell Kernel Help Python 2

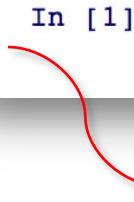
Cell Toolbar: None

# Welcome to the Notebook Technology

This is a markdown cell. You can add LaTex code:

$$\sum_{n=-\infty}^{\infty} |x(n)|^2$$

In [1]: `def thisFunction():  
 return 42`



This is a notebook in Python

Code

Welcome to the Notebook Technology

This is a markdown cell. You can add LaTex code:

$$\sum_{n=-\infty}^{\infty} |x(n)|^2$$

```
In [1]: def thisFunction():
    return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

Code



File Edit View Insert Cell Kernel Help | Python 2 O

Cell Toolbar: None

# Welcome to the Notebook Technology

This is a markdown cell. You can add LaTex code: 
$$\sum_{n=-\infty}^{\infty} |x(n)|^2$$

```
In [1]: def thisFunction():
    return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash
curl roottaasdemo.web.cern.ch/roottaasdemo/SaaSFee.jpg \
> SF.jpg
```

Shell Commands

→ We can invoke commands in the shell...

File Edit View Insert Cell Kernel Help Python 2 O

Cell Toolbar: None

# Welcome to the Notebook Technology

This is a markdown cell. You can add LaTex code: 
$$\sum_{n=-\infty}^{\infty} |x(n)|^2$$

```
In [1]: def thisFunction():
    return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash
curl rootoasdemo.web.cern.ch/rootoasdemo/SaasFee.jpg \
> SF.jpg
```

% Total Time	% Received Time	% Xferd Current	Average Speed Dload	Time Upload	Total
Spent 100	Left 128k	Speed 100 128k 0	0 2731k	0	--:---:--
--:---:--	--:---:--	2787k			

... And capture their output



File Edit View Insert Cell Kernel Help Python 2 O

Cell Toolbar: None

# Welcome to the Notebook Technology

This is a markdown cell. You can add LaTex code: 
$$\sum_{n=-\infty}^{\infty} |x(n)|^2$$

```
In [1]: def thisFunction():
    return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash
curl rootoasdemo.web.cern.ch/rootoasdemo/SaaSFee.jpg \
> SF.jpg
```

% Total Time	% Received Time	% Xferd Current	Average Speed Dload	Time Upload	Total
Spent 100	Left 128k	Speed 100 128k	0 0 2731k	0	--:--:--
--:--:--	--:--:--	2787k			

```
In [4]: from IPython.display import Image
Image(filename=".SF.jpg",width=225)
```





In [3]:

```
%%bash
curl rootaaSdemo.web.cern.ch/rootaaSdemo/SaaSFee.jpg \
> SF.jpg

% Total    % Received % Xferd  Average Speed   Time
Time      Time  Current
                                         Dload  Upload   Total
Spent     Left  Speed
100  128k  100  128k    0       0  2731k       0  --::--::--
--::--::-- --::--::-- 2787k
```

In [4]:

```
from IPython.display import Image
Image(filename=".//SF.jpg",width=225)
```

Out[4]:

**Images**



## In a browser

```
function():
    42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

## Code

```
curl -O https://saasdemo.web.cern.ch/root/aasdemo/SaasFee.jpg \
```

```
> SF.jpg
```

% Total Time	% Received Time	% Xferd Current	Average Speed	Time
Spent 100	Left 128k	Speed 128k	0 0 2787k	0 --::--
--::--	--::--	--::--	2787k	--::--

## Shell Commands

```
In [4]: from IPython.display import Image
Image(filename=".//SF.jpg",width=225)
```

```
Out[4]:
```



## Images

# JupyterLab



- JupyterLab is the next generation of the Jupyter Notebook.
- JupyterLab offers a general framework for interactive computing and data science in the browser, using Python, Julia, R, or one of many other languages.

The screenshot shows the JupyterLab interface with the following components:

- File Browser:** On the left, it lists notebooks like Lorenz.ipynb, lorenz.py, and R.ipynb, along with their modification times.
- Code Editor:** The main area contains a notebook cell [2]:

```
from lorenz import solve_lorenz
interactive(solve_lorenz, sigma=(0.0,50.0), rho=(0.0,50.0))
```
- Output View:** Below the code editor, there's an "Output View" section with sliders for `sigma`, `beta`, and `rho`, and a corresponding Lorenz attractor plot.
- Terminal:** A tab labeled "Terminal 1" is visible at the top.
- Console:** A tab labeled "Console 1" is visible at the top.
- Data:** A tab labeled "Data.ipynb" is visible at the top.
- README:** A tab labeled "README.md" is visible at the top.
- Kernel:** A tab labeled "Python 3 (ipykernel)" is visible at the top.
- Code Editor:** On the right, there's a separate code editor window titled "lorenz.py" containing the following Python code:

```
def solve_lorenz(sigma=10.0, beta=8./3, rho=28.0):
    """Plot a solution to the Lorenz differential equations."""
    fig = plt.figure()
    ax = fig.add_axes([0, 0, 1, 1], projection='3d')
    ax.axis('off')

    # prepare the axes limits
    ax.set_xlim((-25, 25))
    ax.set_ylim((-35, 35))
    ax.set_zlim((5, 55))

def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
    """Compute the time-derivative of a Lorenz system."""
    x, y, z = x_y_z
    return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]

# Choose random starting points, uniformly distributed from -15 to 15
np.random.seed(1)
x0 = -15 + 30 * np.random(N, 3)
```