NUMERICAL SOLUTION AND SIMULATION IN ENGINEERING WITH PYTHON

By:

Ayoob Salari

Chapter 1

Table of Contents

SECTION 1

Python Installation

We use ***Conda*** as the package and environment manager that allows us to execute command line commands at the Anaconda Prompt for Windows OS. For Linux and macOS, we can use terminal to run the commands. ***Anaconda*** is free to download and is available on all types of operating systems. To download anaconda, click the url to the right: https://www.anaconda.com. Anaconda features a powerful graphical user interface that is useful for individuals who wish to begin programming in Python.

1.1 Managing Environment

When we program in Python, we utilize several packages and libraries. Depending on the scope of our project, we may need different versions of Python or libraries. Therefore, it is advised to use a Python virtual environment, which entails creating an environment for each project and install the appropriate version of Python and packages depending on the project's specifications.
We can use the terminal or Anaconda Prompt to create an environment with a specific version of Python

```
conda create –n myenv python=3.9
```

Replace myenv with your environment name. When Conda instructs you to proceed, type y:

```
proceed ([y]/n)?
```

Once the environment has been created, it must be activated to install the necessary libraries based on the requirements of the project.

```
conda activate myenv
```

Replace myenv with your environment name. It should be noted that one can also install the packages during the process of creating the environment. Similarly, to deactivate an environment, type

```
conda deactivate
```

To remove an environment, we can use

```
conda remove --name myenv --all
```

and to verify that the environment was removed, in Anaconda Prompt, run:

```
conda info --envs
```

To further learn how to Share/Restore/Clone/Update an environment, visit Conda document for *managing environments*.

1.2 Python IDEs

To make writing, testing, and debugging simpler, an integrated development environment (IDE) provides features like code completion, code insight, resource management, debugging tools, and more.

Jupyter Notebook

Open-source IDE Jupyter notebook is used to generate and distribute Jupyter documents containing live coding. It is a web-based computational environment where users can participate in real time. Its extensive formatting and user-friendly design make it a popular choice among beginners. Python, Julia, Scala, R, and other prominent data science languages are all supported by the Jupyter notebook.

PyCharm

PyCharm is a specialized Python IDE that includes a broad variety of key Python developer tools that are closely integrated to offer a pleasant environment for effective Python, web, and data science development. It is a great Python IDE, but it might be sluggish to load and the default settings may need to be tweaked for existing projects to work properly.

Visual Studio (VS) Code

VS code is considerably quicker than PyCharm, and it saves everything even if you do not use the *ctrl* + *s* shortcut. When you open VS code, it will immediately detect your Python installation and libraries.

Spyder

Spyder was designed from the ground up as a data science-specific IDE and is included in the Anaconda package management distribution. Those familiar with R or Matlab will find Spyder an excellent IDE for learning Python because of the comparable interfaces. Spyder is one of the best IDEs for Python scientific programming, data science, and machine learning applications.
Spyder, which has a Matlab-like interface and is familiar to the majority of engineers, will serve as our programming IDE for the rest of this tutorial. Like Matlab, Spyder has a console, a variable explorer, and a debugger built in by default.

Launch Spyder While Spyder can be downloaded and installed directly from the *Spyder website*, it is advised for beginners to install it via Anaconda since it comes with everything you need.
After installing Anaconda, you can use the Anaconda Navigator to view all installed applications. Clicking on the Home button and the Launch icon will open the Spyder (Fig. 1).

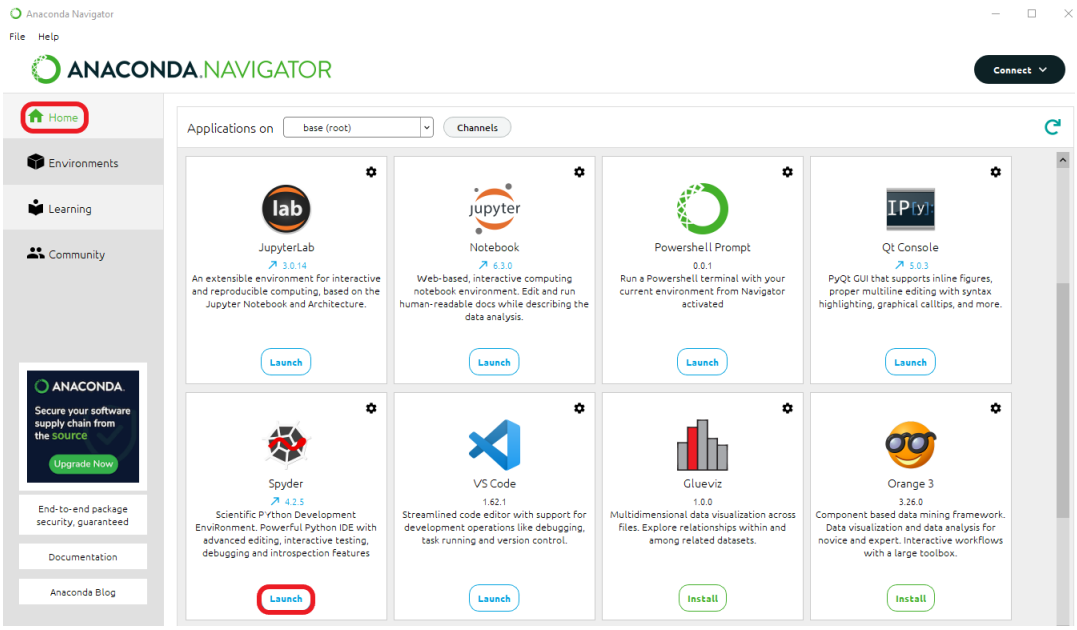

Figure 1: Launching Spyder via Anaconda Navigator

Another way to launch the Spyder in Anaconda is by opening the Anaconda Prompt, activating the environment that you want to work with and typing spyder as follows

```
conda activate myenv
spyder
```

Replace myenv with your environment name.

1.3 Top Python Libraries

Libraries are collections of classes and modules that may be used to execute certain tasks without developing the features from start. Despite the seemingly endless number of Python libraries, there are a few essentials on the list that all programmers should be familiar with. These top libraries are as follows:

Pandas

Python data analysis (Pandas) is a commonly used library in data science and analytics. DataFrame or Series are two-dimensional pandas table objects that contains column names and row labels. Pandas offers a comprehensive range of tools for data exploration, cleaning, and analysis. All types of structured data may be loaded, prepared, manipulated, and analyzed using Pandas.

NumPy

Numerical Python (NumPy) is an array-processing library that handles large matrices, multidimensional data, linear algebra, and Fourier transform and contains fast precompiled functions for compact and faster computations. Other libraries, such as TensorFlow, also utilize NumPy for core tensor computing.

Scikit-learn

One of the most popular machine learning libraries, Scikit-learn contains a wide range of machine learning algorithms, such as clustering and classification. Scikit-learn is used to create machine learning models after cleaning and processing data using Pandas or NumPy.

TensorFlow

TensorFlow is a popular Python neural network library. It employs multidimensional arrays, called tensors, to execute numerous operations on a particular input. Tensorflow enables concurrent training of several neural networks on both the CPU and GPU.

Keras

Deep learning models, such as neural networks, are the primary focus of Keras. Neural networks can be developed quickly and easily thanks to the use of TensorFlow and Theano as the core of Keras. Pre-labeled datasets and pre-trained models can be directly imported and loaded into Keras, making deep learning very simple.

SciPy

Optimization, linear algebra, and statistics are all included in the scientific Python (SciPy) machine learning package. It utilizes NumPy to do efficient numerical calculations and to solve mathematical functions.

Plotly

Plotly is a strong graphing tool that enables users to interact with the visualizations. Numerous unique chart formats are supported by the plotly Python library for a broad variety of statistical, scientific, and 3D applications.

Matplotlib

Matplotlib is a robust visualization and graphical plotting library for Python that can be used in lieu of MATLAB. The Matplotlib APIs can be used to embed plots in graphical user interface (GUI) applications.

PyTorch

By using a graphics processing unit (GPU) to accelerate tensor calculations, PyTorch is one of the most widely used machine learning libraries. In a number of ways, PyTorch beats TensorFlow, and as a result, it has received a great deal of attention lately.