PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

Mohammed Khalid Hossen
PhD Candidate
Dalhousie University, Canada.

DALHOUSIE UNIVERSITY

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

OUTLINES:

- Why we need programming languages for CFD?
- Which programming languages should we learn for CFD?
- Basic Programming for Python.
- Algorithm and Data structures of basic python.
- Coding for function (It is called as in built in a programming language).
- Coding for Matrix Algebra.
- Coding for differencing methods
- Coding for solving equations related to fluid dynamics.
- Finally, COMSOL software.

Mohammed Khalid Hossen

DALHOUSIE UNIVERSITY

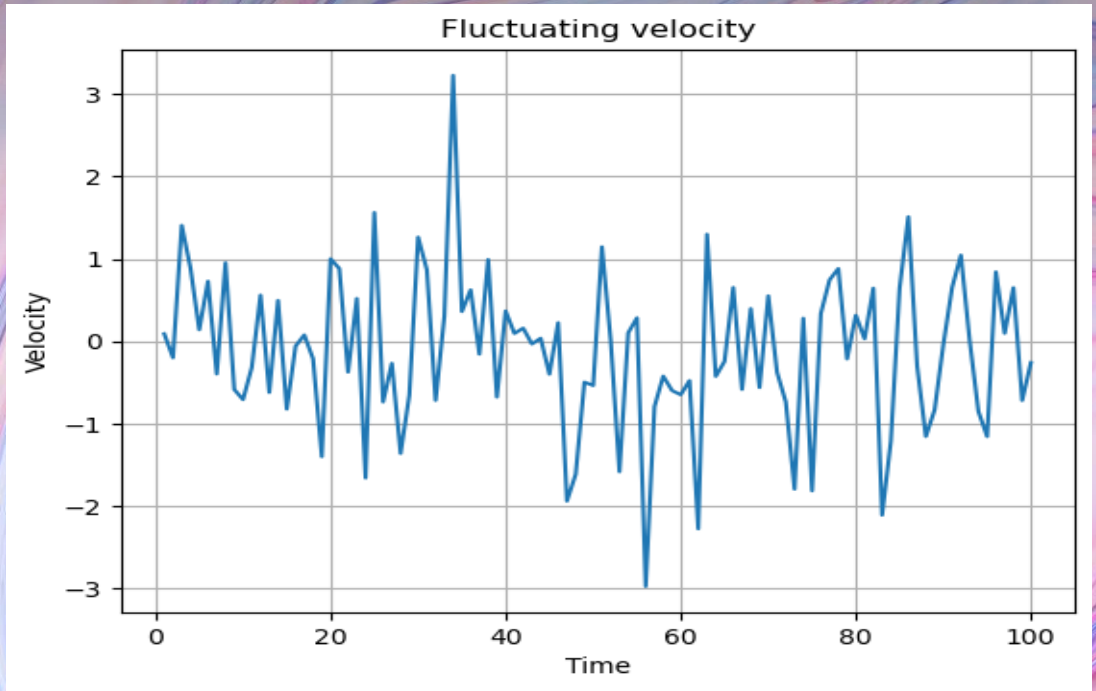
PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑Modelling Fluctuating Velocity

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

N = 100
t = np.arange(1, N + 1)
v = np.random.randn(N)

plt.plot(t, v)
plt.title('Fluctuating velocity')
plt.xlabel('Time')
plt.ylabel('Velocity')
plt.grid(True)
plt.show()
```



Mohammed Khalid Hossen

DALHOUSIE UNIVERSITY

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑ Why we need programming languages for CFD?

Computational Fluid Dynamics (CFD) is a complex field that involves simulating and analyzing the behavior of fluids (such as liquids and gases) using **numerical methods and algorithms**. Programming languages are essential in CFD for several reasons:

1. <u>Numerical Simulations</u>: CFD involves solving complex mathematical equations, such as the Navier-Stokes equations, which describe the behavior of fluids. These equations cannot be solved analytically.

2. <u>Customization</u>: CFD simulations often require custom algorithms and solvers tailored to specific problems or geometries.

Mohammed Khalid Hossen

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

- **Why we need programming languages for CFD?**

3. High Performance: CFD simulations can be computationally intensive, requiring the solution of large systems of equations and the handling of vast amounts of data.

4. Libraries and Frameworks, Portability, Collaboration, Post-processing and Visualization, etc.

Mohammed Khalid Hossen

DALHOUSIE UNIVERSITY

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑ Which programming languages should we learn for CFD?

- When it comes to learning programming languages for Computational Fluid Dynamics (CFD), **the choice depends on your specific goals, preferences, and the type of work** you plan to do in CFD.

- The choice of programming languages for CFD depends on your specific needs, the CFD software you plan to work with, and the computational resources available to you. Learning multiple languages and having a strong understanding of the underlying principles of numerical methods and algorithms will make you a more versatile and effective CFD practitioner.

Mohammed Khalid Hossen

DALHOUSIE UNIVERSITY

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑ Which programming languages should we learn for CFD?

- Here are some programming languages commonly used in CFD:

Mohammed Khalid Hossen

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑ Python Programming Language?

Installation Process of Python Programming Language:

The installation process for Python varies slightly depending on your operating system (e.g., Windows, macOS, Linux) and whether you want to install Python 2.x or Python 3.x.

Mohammed Khalid Hossen

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑ Python Programming Language?

Here are the general steps for installing Python:

1. Visit the Python Website: https://www.python.org/ and click on the "Downloads" tab.

2. Choose the Python Version: Python is available in two main versions: Python 3.x and Python 2.x

3. Download the Installer: Click on the link to download the installer for your operating system. Make sure to select the appropriate installer for your system (Windows, macOS, or Linux).

4. Run the Installer: After downloading the installer, run it.

5. Install Python: Python will be installed in the default location on your system. **Verify the Installation**

Mohammed Khalid Hossen

DALHOUSIE UNIVERSITY

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑ Python Programming Language?

Install a Code Editor or IDE:

- **While Python can be run from the command line, many developers prefer using code editors or integrated development environments (IDEs) for coding. Popular choices include Visual Studio Code, PyCharm, and Jupyter Notebook.**

Let's start Python Programming ☺

Mohammed Khalid Hossen

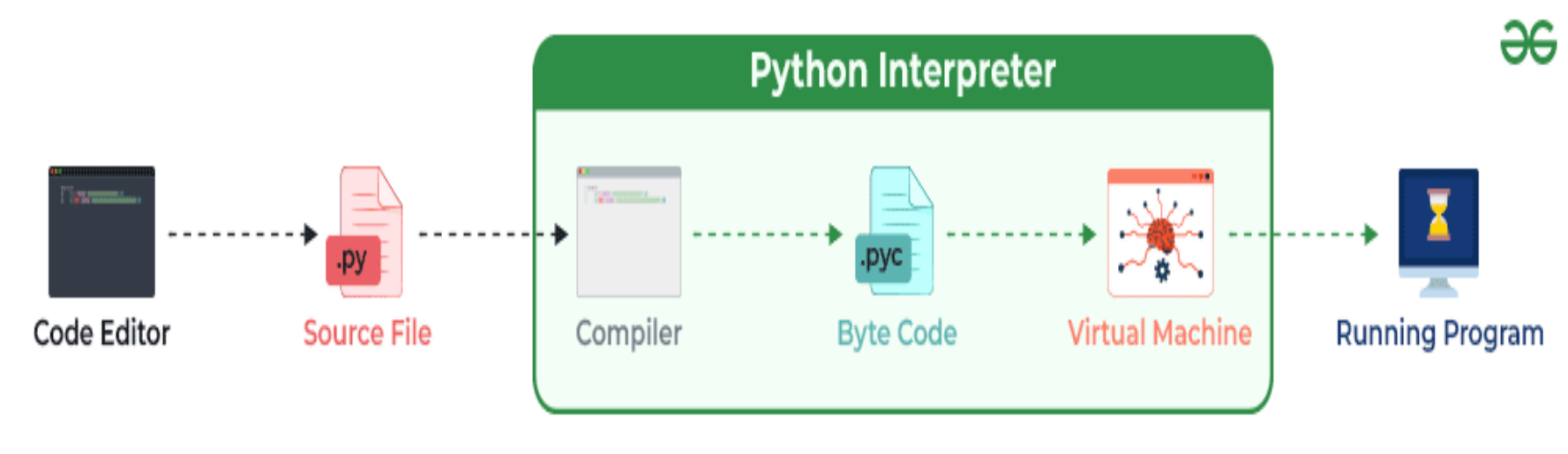
DALHOUSIE UNIVERSITY

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

Mohammed Khalid Hossen

- Python Programming Language?

How Python Internally works:



DALHOUSIE UNIVERSITY

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑ Python Programming Language?

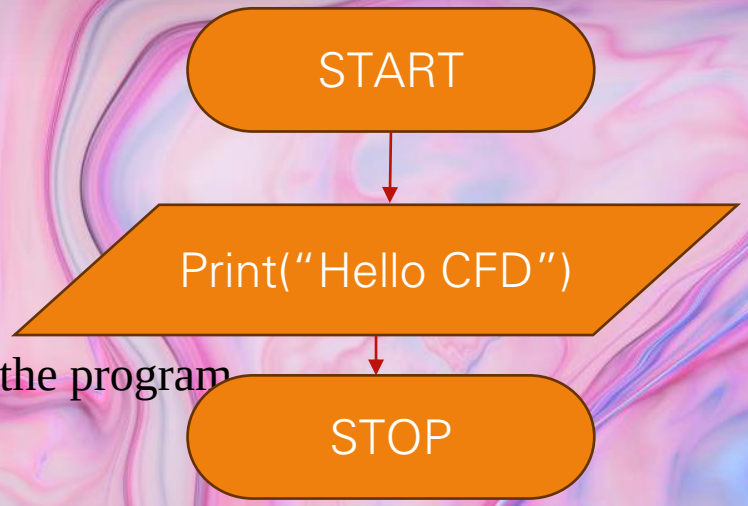
- print("Hello CFD")

Algorithm: An algorithm is a finite, step by step procedure written using the natural language, i.e., generally n English

Algorithm to print Hello World:

1. Start
2. print "Hello CFD"
3. End

Flow Chart: It is a graphical representation of control flow of the program



Mohammed Khalid Hossen

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑ Python Programming Language?

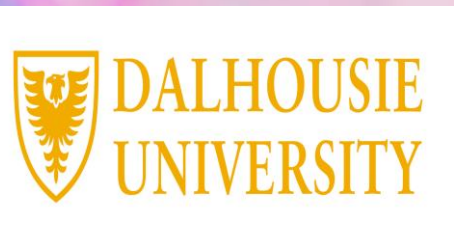
➢ **How works for print("Hello CFD") in python programming?**

1. **Parsing the Code**: The Python interpreter first parses the code to understand its structure and syntax.

2. **Executing the print() Function:** When the interpreter encounters the print() function, it executes it.

3. **Evaluating the Arguments:** Python evaluates the argument inside the print function.

4. **Displaying the Output**: The print() function displays the evaluated argument, "Hello World," on the standard output, which is typically your computer's console or terminal.



Mohammed Khalid Hossen

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑ Python Programming: Data Types

Here are some of the fundamental data types in Python:

1. Integer (int): Represents whole numbers, both positive and negative. For example: **x =42, y = -89.**

2. Float ('float'): Represents floating-point numbers (decimal numbers). For example: **pi = 3.14159 value = -0.5**

3. Boolean ('bool'): Represents either **'True' or 'False'.** Often used for logical operations and control flow. For example: is**_raining = True, is_sunny = False**

4. String ('string'): Represents a sequence of characters enclosed in single ('') or ('' "") quotes. For Example: **name = "John", message = 'Hello, world!'**

5. List('list'): Represents an ordered, mutable (changeable) collection of items. Lists can contain elements of different data types. For example: numbers = [1, 2, 3, 4, 5],
fruits = ["apple", "banana", "cherry"]

Mohammed Khalid Hossen

PROGRAMMING FOR COMPUTTUINAL FLUID DYNAMICS

❑ Python Programming: Data Types

Here are some of the fundamental data types in Python:

6. Tuple (‘tuple’): Similar to lists, but they are immutable (cannot be changed after creation) and are typically used for storing collections of related data. For example: **coordinates = (3, 4)**

rgb_color = (255, 0, 0)

7. Set (‘set’): Represents an unordered collection of unique elements. Sets are mutable (you can add or remove elements) and are useful for various mathematical operations. For example:
unique_numbers = {1, 2, 3, 4, 5}
vowels = {"a", "e", "i", "o", "u"}

8. Dictionary (‘dict’): Represents a collection of key-value pairs. Each key is unique, and you can use it to look up the associated value quickly. For example: **person = {"name": "Alice", "age": 30, "city": "New York"}**

DALHOUSIE UNIVERSITY

Mohammed Khalid Hossen