Computational Science and Scientific Computing Workshop

Data Programming - Python as a scientific computing tool

Elliot S. Menkah, Ph.D
Daniella N. Apeadu

National Institute for Mathematical Sciences
Kwame Nkrumah University of Science and Technology

October 15, 2025

arrays and multidimensional vectors

Handling Arrays & Multidimensional Vectors

Vector Operation

$$
\begin{aligned}
\vec{a} \cdot \vec{b} &= \sum_{i=0}^{N} a_i b_i \\
&= \begin{pmatrix} 20 & -3 & 5 \end{pmatrix} \begin{pmatrix} 15 \\ -2.249 \\ 1 \end{pmatrix} \\
&= 20(15) - 3(-2.249) + 5(1) \\
&= 300 + 6.747 + 5 \\
&= 311.747
\end{aligned}
$$

Handling Arrays & Multidimensional Vectors

Multidimensional Arrays

$$\begin{bmatrix} 20 & 15 & 10 & 45 \\ -3 & -2.249 & 7 & 1.751 \\ 5 & 1 & 3 & 9 \end{bmatrix} = \begin{pmatrix} 20 \\ -3 \\ 5 \end{pmatrix} \begin{pmatrix} 15 \\ -2.249 \\ 1 \end{pmatrix} \begin{pmatrix} 10 \\ 7 \\ 3 \end{pmatrix} \begin{pmatrix} 45 \\ 1.751 \\ 9 \end{pmatrix}$$

file I/O, exceptions and assertions

File I/O

keyword: **open**

```
>>> fh = open("demofile.txt", "a")
>>> fh.write("My data file \n")
>>> fh.write("Results: %d", res)
>>> fh.close

```

Exceptions and Assertions

This is a way to handle expected and unexpected errors.

1. Exceptions Handling
2. Assertion

```
try:
# Runs First
< code >
except:
# Runs if exceptions occurs in try block
< code >
else:
# Executes if try block succeeds.
< code >
finally:
# This code always runs executes.
< code >

```

Exceptions and Assertions

Exception Example

```
def read_file(path):
""" Return the content of a file at path"""
try:
fh = open(path, mode="rb")
data = f.read()
return data
except FileNotFoundError as err:
raise
else:
fh.close
finally:
print("Leaving file read routine")

```

Python Basics - End

End of Basics.
Questions ?
Review

Numerical and Scientific Python

Numerical and Scientific Python
Numpy and Scipy libraries

Numerical Python - NumPy

Arrays could be made from:

1. Python list or tuples
2. Using functions that are dedicated to generating numpy arrays, such as *arange*, *linspace*, etc.
3. Reading data from files

```
from numpy import as np
v = array([1,2,3,4])
-----
[1,2,3,4]

```

```
M = np.array([[1, 2], [3, 4]])
-----
array([[1, 2],
[3, 4]])

```

Exercises 4

Using the python programming language, write a code that implements the solution or finds the roots of the non-linear equation:

$3x^2 + 2x + 1 = 0$

using the

1. Bisection Method
2. Newton-Raphson's Method
3. Secant Method

as separate functions.

Classes

Classes in Python are defined by the keyword **class**

```
>>> class myfunctions:
...
...     def add(x):
...         res = x + 2
...         return res
>>>
>>> yy = myfunctions.add(7)
>>> yy
9

```

End of talk

Thank you