Com S // Cpr E // Math 5250

Numerical Analysis of High-Performance Computing

Instructor: Songting Luo

Lecture 6: "Main Program" & Matplotlib Demo

Outline

1. “Main Program”
2. Simple File I/O
3. Matplotlib Demo

“Main Program”

“Main Program”

Python modules often end with a section that looks like:

```
if __name__ == "__main__":

        # some code
```

This code is **not** executed if the file is imported as a module, only if it is run as a script, e.g., by ...

From Unix command line:

```
$ python3 filename.py
```

Inside `python`:

```
>>> execfile("filename.py");  #python 2
>>> exec(open("filename.py").read()); #python 3
```

“Main Program”

Python modules often end with a section that looks like:

```
if __name__ == "__main__":

        # some code
```

This code is **not** executed if the file is imported as a module, only if it is run as a script, e.g., by ...

From Unix command line:

```
$ python3 filename.py
```

Inside `python`:

```
>>> execfile("filename.py");  #python 2
>>> exec(open("filename.py").read()); #python 3
```

Quick demo . . .

Simple File I/O

Writing to a file

```
def generate_input_file():
    '''
    This function creates a file called
    input.data. This file will have a
    random number of floats each of which
    is a random number between 0.0 and 1000.0
    '''

    import random as rand
    kmax = rand.randint(1,1000)

    fid = open('input.data', 'w')
    for k in range(0,kmax):
        value = rand.uniform(0.0, 1000.0)
        fid.write("%12.6e" % value)
        fid.write("\n");
    fid.close()

if __name__ == "__main__":
    generate_input_file()
```

Reading a file and storing as NumPy array (Part 1)

```
def read_input_file():
    '''
    This function reads the file "input.data"
    and stores the results in the NumPy array A.
    '''

    # First, figure out how many floats there are
    fid = open('input.data', 'r')
    kmax = 0;
    while True:
        line = fid.readline()
        if not line: break
        kmax = kmax+1
    fid.close()
```

Reading a file and storing as NumPy array (Part 2)

```
    import numpy as np
    import string as str

    # Second, read-in all the floats
    A = np.zeros(kmax,dtype=float)
    fid = open('input.data', 'r')
    for k in range(0,kmax):
        linestring = fid.readline()
        linelist   = linestring.split()
        A[k]       = np.float(linelist[0])
    fid.close()

    # Third, return the result
    return A;

if __name__ == "__main__":
    A = read_input_file()
    print( A.shape[0] )
```

Lab assignment

- Develop a module **mylinalg.py** that contains the **GaussElimination(A,b)** function produced in last assignment.
- Develop a function **LeastSquareApprox(x,f,n)** that find the least square approximation of data {x, f} by a polynomial of degree $\leq n$. The output is the coefficient of the polynomial. Include it in the module **mylinalg.py**.
- Application: find the polynomial $p(x)$ of degree ≤ 5 such that it approximates $f(x) = \cos(x)$ at the nodes $linspace(-\pi, \pi, 51)$ in the least square sense.
- Include the **"main program"** in **mylinalg.py** for the application.
- Run **mylinalg.py** as script
- Update Git repository
- Submit both codes and results (screenshot and figure)

Matplotlib Demo