Physics 40: Laboratory Nineteen

Tuesday-Thursday, June 02-04, 2020

This is the last lab of the quarter. I will be here on Thursday for those who do not finish today.

Today's Goals:

numpy
Random numbers, random walks in python.
Python for the $d = 1$ diffusion equation.
Python for the $d = 2$ Laplace equation.

[**0**] *NumPy* is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. For details:

`https://docs.scipy.org/doc/numpy-1.13.0/user/whatisnumpy.html`

[**1**] Here's a simple script to get N random numbers:

```
# PYTHON ADD-ONS FOR ARRAYS, LINEAR ALGEBRA, STATS, RANDOM NUMBERS, ..
import numpy
import random

# INPUT N
N = input('Enter N:  \n')

# GET N RANDOM NUMBERS UNIFORM ON (0,1)
for i in range (0,N):
    print(i,numpy.random.rand())
```

[**HW9-7**] Modify the script to compute the first four moments. See Lab 9.

Here is a python script to solve the diffusion equation in one dimension. Reviewing Labs 8 and 9 might be useful!

```
# THIS IS NEEDED TO USE ARRAYS IN PYTHON:
from array import *
# THIS IS NEEDED TO MAKE PLOTS IN PYTHON:
import matplotlib.pyplot as plt

# INPUTS
Nt=input('Enter Nt:  \n')
dt=input('Enter dt:  \n')
Nx=input('Enter Nx:  \n')
dx=input('Enter dx:  \n')
D =input('Enter D :  \n')

# DEFINE USEFUL CONSTANTS
Ddtodx2= D*dt/(dx*dx)

# START OFF DENSITY ARRAYS
rho =array('f', [0.0])
for ix in range (1,Nx):
    rho.append(0.0)
rho.insert(Nx/2,1.0)

newrho =array('f', [0.0])
for ix in range (1,Nx):
    newrho.append(0.0)
newrho.insert(Nx/2,1.0)

# LOOP OVER TIME STEPS
for it in range (1,Nt):
    t=dt*it
#   LOOP OVER SPATIAL 'BOXES'
    for ix in range (1,Nx):
        newrho[ix]=rho[ix]+Ddtodx2*(rho[ix-1]-2.*rho[ix]+rho[ix+1])
    for ix in range (1,Nx):
        rho[ix]=newrho[ix]

# OUTPUT FINAL DENSITY
for ix in range (0,Nx+1):
    x=dx*ix
    print(x,rho[ix])
```

Type it in and see that it works. How does the execution time of this python script compare to your C code? I got 65 seconds for python, less than one second for C. This illustrates one of the down-sides of python. It can be much slower than a compilable language like C.

[**HW9-8**] Modify the code above to make a plot of the final density. Run it for $N_t = 10^5, dt = 0.0001, N_x = 10^3, dx = 0.01, D = 0.02$ and hand in your plot.
Extra Credit: Can you figure out a way to read in, and then plot, the analytic solution together with your numeric solution? The data file is on the course website, along with the code which generates it. What happens if you try to increase dt by using:
$N_t = 2 \times 10^4, dt = 0.0005$?
$N_t = 10^4, dt = 0.001$?
$N_t = 2 \times 10^3, dt = 0.005$?

[**HW9-9**] Write a python script to solve the Laplace equation in two dimensions. Lab 12 might be useful!