Computational Analysis of Physical Systems (Lecture 3)

Operations on arrays

Plotting commands

Motion in two dimensions

Arrays in Python

- For numerical computation we need:
  - VECTORS
  - MATRICES

Vectors-1 (Native)

```
a=[1,2,3,4,5]
                    [1, 2, 3, 4, 5]
a[2]
                    3
a.append(6)
                    [1, 2, 3, 4, 5, 6]
a.insert(0,7)
                    [7, 1, 2, 3, 4, 5, 6]
```

Vectors-2 (Native)

a[2]=8

[7, 1, 8, 3, 4, 5, 6]

a[1:3]

[1, 8]

a[1:4]

[1, 8, 3]

Vectors-3 (Native)

print a

[7, 1, 8, 3, 4, 5, 6]

len(a)

7

2*a

[7, 1, 8, 3, 4, 5, 6, 7, 1, 8, 3, 4, 5, 6]

Matrices (Native)

```
m=[[1,2,3],[4,5,6],[7,8,9]]
            [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
len(m)
            3
m[1]
            [4, 5, 6]
m[1][1]
            5
```

NumPy module-1

```
from numpy import *
a= array([[1.,2.,3.],[4.,5.,6.]])
print a
```

```
[[ 1.  2.  3.]
 [ 4.  5.  6.]]
```

NumPy module-2

Number of dimensions:

ndim(a)

2

shape(a)

(2, 3)

a[1,2]

6.0

NumPy module-3

```
a[1,:]
            array([ 4.,  5.,  6.])
a[1]
            array([ 4.,  5.,  6.])
a[:,1]
            array([ 2.,  5.])
```

NumPy module-4

a.transpose()

```
array([[ 1.,  4.],
       [ 2.,  5.],
       [ 3.,  6.]])
```

a.T

```
array([[ 1.,  4.],
       [ 2.,  5.],
       [ 3.,  6.]])
```

NumPy module-5

```
a.conj().transpose()
a.conj().T
        array([[ 1.,  4.],
               [ 2.,  5.],
               [ 3.,  6.]])
```

NumPy module-6

```
a=array([[ 1.,  2.,  3.],[ 4.,  5.,  6.]])
b=array([[0.1,0.2],[0.3,0.4],[0.5,0.6]])
dot(a,b)
                array([[ 2.2,  2.8],
                       [ 4.9,  6.4]])
a*a
                array([[  1.,   4.,   9.],
                       [ 16.,  25.,  36.]])
```

NumPy module-7

a**2

```
array([[  1.,   4.,   9.],
       [ 16.,  25.,  36.]])
```

power(a,2)

```
array([[  1.,   4.,   9.],
       [ 16.,  25.,  36.]])
```

NumPy module-8

```
v=arange(1,11)
        array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
v=arange(10)
        array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
arange(1,4)[:, newaxis]
        array([[1],
               [2],
               [3]])
```

NumPy module-9

```
zeros((3,4))
            array([[ 0.,  0.,  0.,  0.],
                   [ 0.,  0.,  0.,  0.],
                   [ 0.,  0.,  0.,  0.]])
ones((3,4))
            array([[ 1.,  1.,  1.,  1.],
                   [ 1.,  1.,  1.,  1.],
                   [ 1.,  1.,  1.,  1.]])
```

NumPy module-10

```
eye(3)
            array([[ 1.,  0.,  0.],
                   [ 0.,  1.,  0.],
                   [ 0.,  0.,  1.]])
m=array([[9,8,7],[6,5,4],[3,2,1]])
diag(m)
            array([9, 5, 1])
```

NumPy module-11

```
numpy.random.rand(2,2)
   array([[ 0.89391809,  0.16193813],
          [ 0.31479988,  0.11979686]])
linspace(1,3,4)
   array([ 1., 1.66666667, 2.33333333, 3.])
```

NumPy module-12

a.max()

6.0

a.min()

1.0

linalg.inv(m)

array([[-4.50359963e+15,   9.00719925e+15, -4.50359963e+15],
  [9.00719925e+15, -1.80143985e+16, 9.00719925e+15],
  [-4.50359963e+15,   9.00719925e+15, -4.50359963e+15]])

NumPy module-13

```
print m
        array([[9, 8, 7],
               [6, 5, 4],
               [3, 2, 1]])
v=array([1,2,3])
linalg.solve(m,v)
        array([ 11.66666667, -20.,  8.])
```

Plotting (Matplotlib)-1

```
from pylab import *
x=arange(0,5*pi,0.01)
y=2*sin(x)
plot(x,y)
show()
```

Plotting (Matplotlib)-2

```
from pylab import *
x=arange(0,5*pi,0.01)
y=2*sin(x)

xlabel('x-axis')
ylabel('y-axis')
plot(x,y)
title('Plot of $y=\sin (x)$')
show()
```

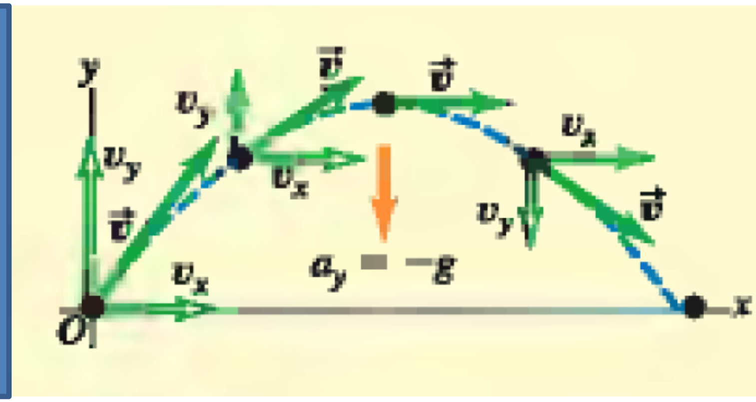
Motion in two dimensions without air resistance

$$x = (v_0 \cos\alpha_0)t$$

$$y = (v_0 \sin\alpha_0)t - \frac{1}{2}gt^2$$

$$v_x = v_0 \cos\alpha_0$$

$$v_y = v_0 \sin\alpha_0 - gt$$



Ask from the user: v0, alpha0, g

Take t-increment as 0.01 seconds.

Stop when y=0.

1. Plot the motion of the particle

2. Find the maximum values of x and y

Solution

```
from math import radians,sin,cos
from pylab import plot,xlabel,ylabel,title,show

v0=input("Enter v0 (m/s)... ")
alpha0=input("Enter alpha0 (degrees)... ")
g=input("Enter g (m/s^2)... ")

radalpha0=radians(alpha0)
t_inc=0.01
t=0.
i=0
x=[]
y=[]
x.append(v0*cos(radalpha0)*t)
y.append(v0*sin(radalpha0)*t-0.5*g*t*t)

while y[i]>=0:
   i=i+1
   t=t+t_inc
   x.append(v0*cos(radalpha0)*t)
   y.append(v0*sin(radalpha0)*t-0.5*g*t*t)

xlabel('x')
ylabel('y')
plot(x,y)
title('Motion in two dimensions')
show()
```