tutorial_numerico

Dario Mitnik

May 19, 2015

Part I

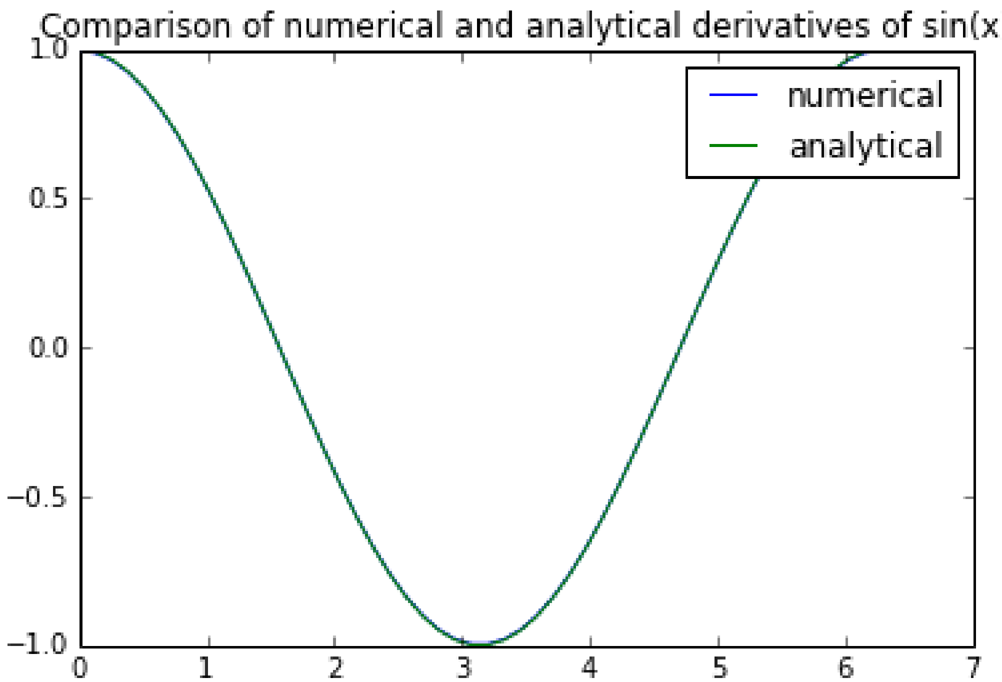
Cálculo Numérico: Derivadas

In [1]:

```
# Función numerica para la derivada primera
# Diferencias Finitas

def dy_diff(y,x):
    "Finite difference derivative of the function f"
    n = len(y)
    d = zeros(n,'d') # assume double

    # Use centered differences for the interior points,
    # one-sided differences for the ends
    for i in range(1,n-1):
        d[i] = (y[i+1]-y[i-1])/(x[i+1]-x[i-1])
        d[0] = (y[1]-y[0])/(x[1]-x[0])
        d[n-1] = (y[n-1]-y[n-2])/(x[n-1]-x[n-2])
    return d
```

In [2]:

```
from numpy import pi, sin, cos, linspace, zeros
from matplotlib.pyplot import plot, title, legend, show
%matplotlib inline

x = linspace(0,2*pi)
dsin = dy_diff(sin(x),x)
plot(x,dsin,label='numerical')
plot(x,cos(x),label='analytical')
title("Comparison of numerical and analytical derivatives of sin(x) ")
legend()
show()
```

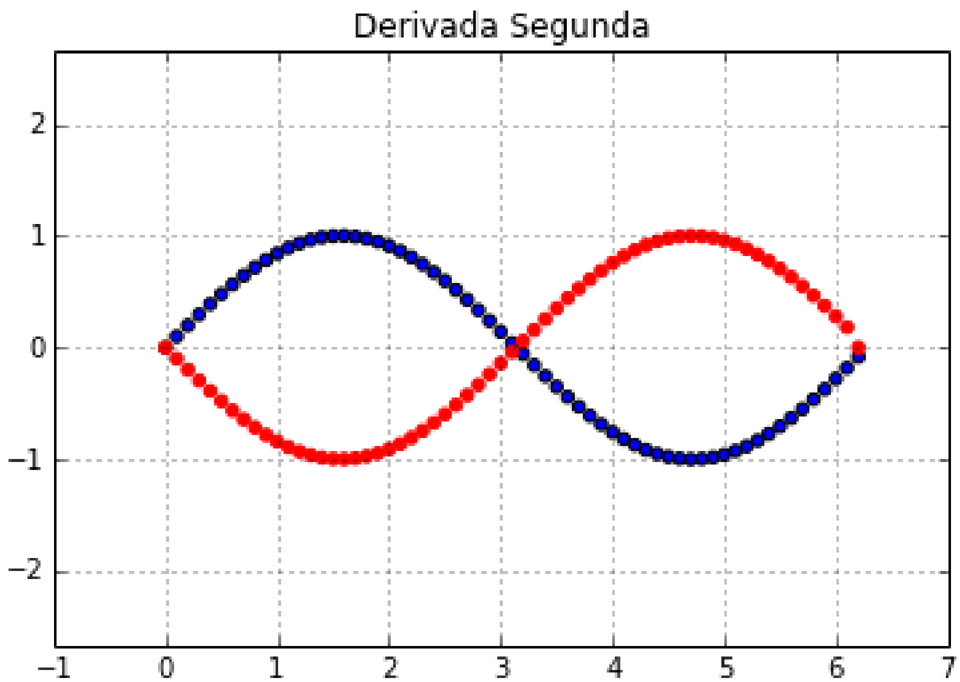


In [4]:

```
# Función Numérica para la derivada segunda
# Diferencias Finitas

def d2y_diff(y, h):
  N = len(y)
  d2y = zeros(N)
  for k in range(1, N - 1):
   d2y[k] = (y[k+1] - 2*y[k] + y[k-1])/(h**2)
  return d2y
```

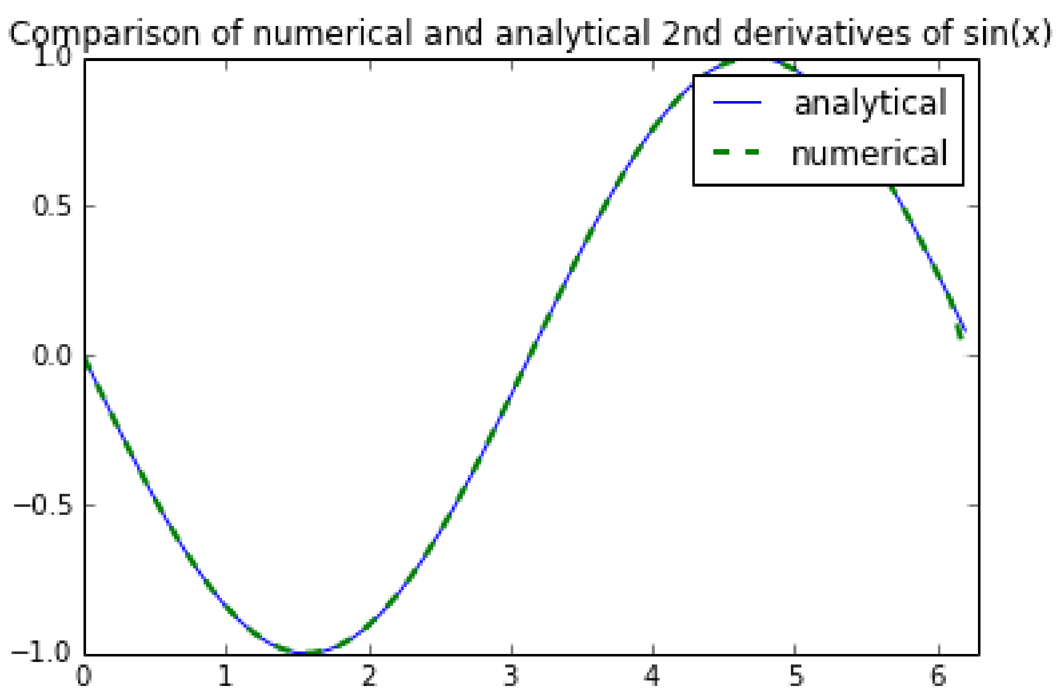
In [5]:

```
from numpy import pi,arange,sin,zeros
from pylab import scatter,grid,title,plot,axis,show

# ver los gráficos en la misma ventana
%matplotlib inline

a,b,h = 0, 2*pi, 0.1
x = arange(a,b,h)
y = sin(x)
d2y = d2y_diff(y, h)

scatter(x,y)
scatter(x,d2y, color='r')
axis('equal')
grid()
title('Derivada Segunda')
show()
```



In [6]:

```
from matplotlib.pyplot import axes,legend

plot(x,-sin(x), label='analytical')
plot(x,d2y,lw=2,linestyle='dashed', label='numerical')
title("Comparison of numerical and analytical 2nd derivatives of sin(x) ")
axis([a,b,-1,1])
legend()
show()
```



1 Programa Simple: Ecuación de 2do grado

In [7]:
```
#Ecuación de Segundo Grado en Python
import pylab

print ('Modelo de la ecuacion  a*x^2 + b*x + c = 0')
a = float(input('Valor de a: '))
b = float(input('Valor de b: '))
c = float(input('Valor de c: '))

#calculando  el discriminante
delta = float(b**2-4*a*c)

if delta < 0:
    print('Ecuacion no tiene solucion real')

elif delta ==0:
    s == float(-b/2*a)
    print ('Solucion unica: ',s)
else:
    x1 = float((-b+((b**2 - 4*a*c))**0.5)/(2*a))
    x2 = float((-b-((b**2 - 4*a*c))**0.5)/(2*a))
    print('Las soluciones son las siguientes')
    print('X1: ',x1)
    print('X2: ',x2)
```

```
Modelo de la ecuacion  a*x^2 + b*x + c = 0
Valor de a: -1
Valor de b: 3
Valor de c: 2
Las soluciones son las siguientes
('X1: ', -0.5615528128088303)
('X2: ', 3.5615528128088303)
```

2 Gráfico de la Solución

In [8]:
```
import matplotlib.pyplot as plt
import math
import numpy as np

# plot in separate window
%matplotlib qt

x = np.arange(-10,10,0.5)
y = a*x**2 + b*x + c
plt.plot(x,y)
plt.title('a*x^2 + b*x + c = 0')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

In [ ]:
```

```