Solución Lab 6
FISI 6510

Guillermo Fidalgo

25 de febrero de 2021

Exercise 4.3: Calculating derivatives

Suppose we have a function $f(x)$ and we want to calculate its derivative at a point x. We can do that with pencil and paper if we know the mathematical form of the function, or we can do it on the computer by making use of the definition of the derivative:

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{\delta \to 0} \frac{f(x+\delta) - f(x)}{\delta}.$$

On the computer we can't actually take the limit as δ goes to zero, but we can get a reasonable approximation just by making δ small.

a) Write a program that defines a function `f(x)` returning the value $x(x-1)$, then calculates the derivative of the function at the point $x = 1$ using the formula above with $\delta = 10^{-2}$. Calculate the true value of the same derivative analytically and compare with the answer your program gives. The two will not agree perfectly. Why not?

b) Repeat the calculation for $\delta = 10^{-4}$, 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}, and 10^{-14}. You should see that the accuracy of the calculation initially gets better as δ gets smaller, but then gets worse again. Why is this?

We will look at numerical derivatives in more detail in Section 5.10, where we will study techniques for dealing with these issues and maximizing the accuracy of our calculations.

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "sans-serif",
    "font.serif": ["Palatino"],
})
```

a)

[2]:
```
def f(x):
    return x*(x-1)
x=1
delta=1e-2
dydx=(f(x+delta)-f(x))/delta
print(dydx)
```

```
1.010000000000001
```

El valor analítico de la derivada de la función $f(x) = x(x-1)$ evaluado en x=1 es

$$f'(x) = \frac{d}{dx}(x^2 - x) = 2x - 1$$

$$\boxed{f'(1) = 2(1) - 1 = 1}$$

Si comparamos esto dos números obtenemos que están a 1 % de diferencia

[3]:
```
abs(1-1.010000000000001)*100
```

[3]:
```
1.0000000000000897
```

Estos no coninciden porque por la definición de límites de la derivada estamos insertando un número finito en la $\delta = 0.01$ y no se está tomando el límite cuando $\delta \to 0$. En teoría esto produce el resultado exacto pero numéricamente, por limitaciones de la computadora no se puede hacer $\delta = 0$.

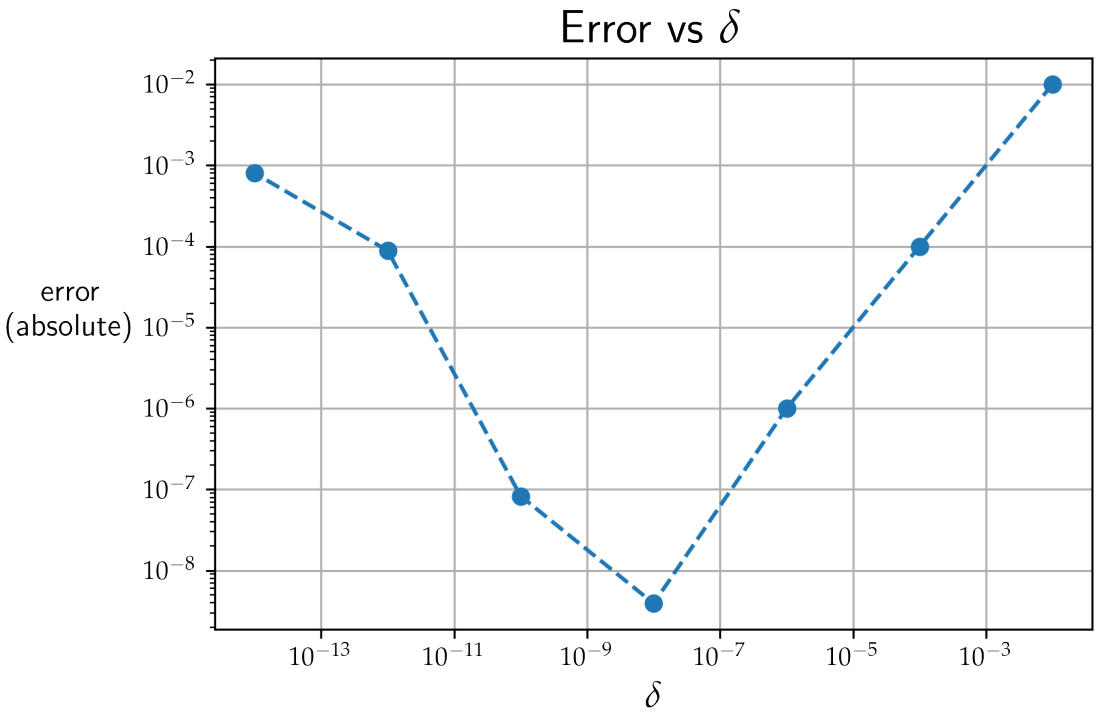
b)

[4]:
```
delta=np.array([1e-2,1e-4,1e-6,1e-8,1e-10,1e-12,1e-14])
dydx=(f(x+delta)-f(x))/delta
err=abs(dydx-1)
for i in dydx:
    print(" dy/dx \t\t\t\t error (absoluto) \t\t error (porcentual) ")
    print(" {} \t \t {} \t\t  {:%}".format(i,abs(i-1),abs(i-1)))
    print('')
# print(err)
```

```
 dy/dx                          error (absoluto)               error
(porcentual)
 1.010000000000001              0.010000000000000897            1.000000%

 dy/dx                          error (absoluto)               error
(porcentual)
 1.0000999999998899             9.999999988985486e-05           0.010000%

 dy/dx                          error (absoluto)               error
```

```
(porcentual)
 1.0000009999177333              9.99917733279787e-07             0.000100%

 dy/dx                           error (absoluto)                error
(porcentual)
 1.0000000039225287              3.922528746258536e-09            0.000000%

 dy/dx                           error (absoluto)                error
(porcentual)
 1.000000082840371               8.284037100736441e-08            0.000008%

 dy/dx                           error (absoluto)                error
(porcentual)
 1.0000889005833413              8.890058334132256e-05            0.008890%

 dy/dx                           error (absoluto)                error
(porcentual)
 0.9992007221626509              0.0007992778373491216            0.079928%
```

[5]:
```
plt.figure(dpi=290)
plt.plot(delta,err,'o--')
plt.yscale('log')
plt.xscale('log')
plt.title("Error vs $\delta$",size=18)
plt.grid()
plt.xlabel('$\delta$',size=14)
plt.ylabel("error \n(absolute)",rotation =0,labelpad=28,size=12)
plt.show()
```



La razón por la que se observa este comportamiento es por que se llegan a las límitaciones de las computadoras en cuanto a su precisión finita. Cuando $\delta < 10^{-8}$ se está tomando restas entre número muy cercanos y se divide por un número de orden menor a 10^{-8}.