Optimization for Big Data - Optimization with Python

Summary

The goal of this first session is to use Python on practical problems of optimization. You are invited to modify your parameters in the code and try to understand what happens with the output of the software.

1 Using Python

Python is a powerful (and free) software useful in nowadays data-science problems. Instead of providing a comprehensive introduction or a complete course on this software, we will learn how to use it through practical examples of mathematical questions and their resolutions with Python.

1.1 Workspace

The workspace is divided into two main parts (see Figure 1).

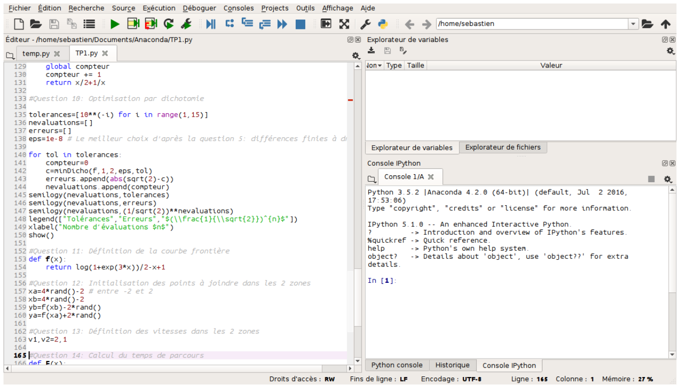

FIGURE 1 – Example of the Spyder workspace.

The left hand side is the Editor useful for program development, or scripts. The right hand side is the console used for the execution of the lines of code. It may be a good idea to write your code on a script file and copy/paste/execute these lines in the console on the right.

1.2 Package

At the very beginning of the Python session, it may be useful to launch the following code. In particular, it makes it possible to use the numpy and matplotlib.

```
In [1]: %pylab inline
```

1.3 Pay a specific attention to :

- How to draw some graphics.
- How to use log/log scales, semi-log scales.
- How to draw some graphs on the same figure (and on different figures).
- How to use while, if and for (have a look at the indentation !).
- How to define some variables (local/global) an re-use them.
- How to handle vectors.

2 Convergence order of sequences

2.1 First sequence

We define the following sequence

$$x_0 = 1 \qquad \text{and} \qquad \forall n \geq 1 : x_{n+1} = x_n\left(1 - x_n/2\right) + 1.$$

Compute with Python the first terms of the sequence (until $N = 10$ for example). Produce a graphical illustration of the convergence and of the order of convergence. For this purpose, you will certainly use the function semilogy.

```
x=1.
list=[x]
for i in range(10):
    x=x*(1.-x/2.)+1.
    list.append(x)
```

```
semilogy(abs(array(list)-sqrt(2)))
show()
```

The graphical output is shown in Figure 2.

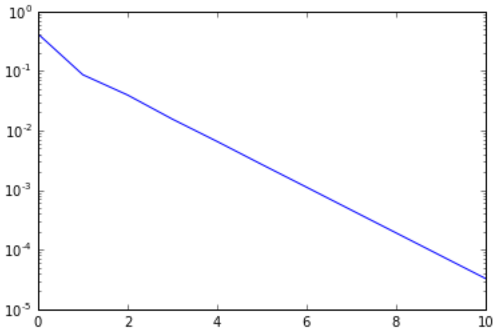

FIGURE 2 – Behavior of the first sequence $(x_n)_{n\geq 1}$.

What is the convergence rate of $(x_n)_{n\geq 1}$? Why is it called "linear" ?

2.2 Second sequence

We define the following sequence

$$y_0 = 1 \qquad \text{and} \qquad \forall n \geq 1 : y_{n+1} = \frac{y_n}{2} + \frac{1}{y_n}.$$

With another way to use the vectors. Have a look at the first element of the vector : position 0 !

```
y=1.
tab=zeros(11)
tab[0]=y
for i in range(1,11):
    y=y/2.+1./y
    tab[i]=y
err2=abs(tab-sqrt(2))
```

```
semilogy(err2)
show()
semilogy(abs(log(err2)))
show()
```

Same questions...

2.3 Third sequence

We define the following sequence

$$z_0 = 2 \quad z_1 = 1 \quad \text{and} \quad \forall n \geq 1 : z_{n+1} = \frac{z_n z_{n-1} + 2}{z_n + z_{n-1}}.$$

Do it alone ! Build the vector err3.

Is the convergence rate still linear ? Explain why it is faster. You can have a look at the Wikipedia webpage on $\sqrt{2}$ for more insight on the origin of these sequences.

3 A first optimization algorithm

In this course, we will often be interested in gradient methods. In some cases, an explicit analytical expression of the gradient is available, so that it can be used directly in the code. However, in some cases, the computation of the gradient is painful it is possible to use an estimation of the derivatives with a numerical method. The two (preliminary) parts explain how we can estimate the derivatives of a smooth function.

3.1 Finite difference method

3.1.1 Theoretical point of view

Assume that f is a real function of class $\mathcal{C}$ with $\|f"\|_\infty \leq C\|f\|_\infty$ where C is a constant (of order 1). We also assume that $\hat{f}$ is an η-approximation of f :

$$\|\hat{f} - f\|_\infty \leq \eta.$$

Show that

$$\left| \frac{\hat{f}(x+\epsilon) - \hat{f}(x)}{\epsilon} - f'(x) \right| \leq \left[\frac{2\eta}{\epsilon} + \frac{C\epsilon}{2} \right] \|f\|_\infty$$

What is the optimal choice of ϵ to obtain a "good" approximation of the derivative of f while using only approximate oracle values of f ?

3.1.2 With Python

- Write a **function** that takes in argument f, x, ϵ and that computes the finite difference estimate of $f'(x)$.

```
def diff_finite(f,eps,x):
    return (f(x+eps)-f(x))/eps
```

- Locate the optimal value of ϵ to estimate $\sin'(1)$ with the finite difference method and keep this value in mind.

3.2 Centered difference method

3.2.1 Theoretical point of view

Assume that f is a real function of class $\mathcal{C}$ with $\|f^{(3)}\|_\infty \leq C\|f\|_\infty$ where C is a constant (of order 1). We also assume that $\hat{f}$ is an η-approximation of f :

$$\|\hat{f} - f\|_\infty \leq \eta.$$

Show that

$$\left|\frac{\hat{f}(x+\epsilon) - \hat{f}(x-\epsilon)}{2\epsilon} - f'(x)\right| \leq \left[\frac{\eta}{\epsilon} + \frac{C\epsilon^2}{6}\right]\|f\|_\infty$$

What is the optimal choice of ϵ to obtain a "good" approximation of the derivative of f while using only approximate oracle values of f ?

3.2.2 With Python

- Write a **function** that takes in argument f, x, ϵ and that computes the centered difference to estimate $f'(x)$.
- Locate the optimal value of ϵ to estimate $\sin'(1)$ with the centered difference method and keep this value in mind.
- Compare this result with the one obtained using the finite difference method.

3.3 Optimization with the dichotomic search

The dichotomic method is a very simple and efficient method for solving the minimization of f as soon as we know that f is decreasing and then increasing on a segment $[a, b]$.

Algorithm 1 Dichotomic minimum search

Input Function f. Tolerance η. Interval bounds $a \leq b$.
While $b - a > \eta$
If $\hat{f}'(\frac{a+b}{2}) < 0$
Then $[a, b] \longrightarrow [\frac{a+b}{2}, b]$.
Else $[a, b] \longrightarrow [a, \frac{a+b}{2}]$
Output : $[a, b]$

- Write a program that minimizes a function (arguments : f, ϵ, η) with an approximation of the derivatives that uses the centered difference method. η represents the accuracy we have chosen to compute the approximation of the minimizer.
- Minimize the function $x \longrightarrow \frac{x}{2} + \frac{1}{x}$.
- Illustrate the convergence rate of the method with a figure .

4 Application : finding the shortest path in two fluids

We are interested in solving the following minimization problem. We want to find the shortest path between two points in an inhomogeneous domain of $\mathbb{R}^2$. The domain takes the form $\Omega = \Omega_1 \cup \Omega_2$ where

$$\Omega_1 := \{(x, y) : y \geq f(x)\},$$

and

$$\Omega_2 := \{(x, y) : y \leq f(x)\}.$$

Here, f is any convex function between -2 and 2. For example :

$$f(x) = x^2 \quad f(x) = e^x \quad f(x) = \log(1+e^x) \quad f(x) = \log(1+e^{3x}) - x + 1 \ldots.$$

Figure 3 shows an example.

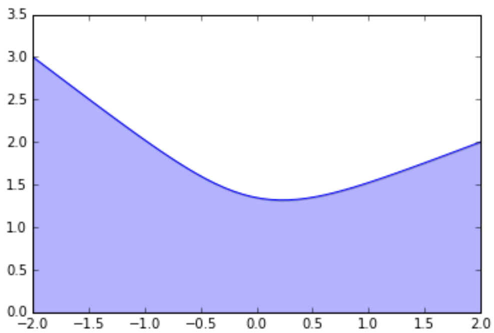

FIGURE 3 – Example of two sub-domains Z_1 (in white) and Z_2 (in blue)

We assume that the sub-domains are inhomogeneous : the travel speed is v_1 in Z_1 and v_2 in Z_2.

- We will assume that the first point z_1 is in Z_1 and the second point z_2 in Z_2. Compute the time needed to go from z_1 to z_2.
- Draw the initial and ending points at random :

```
xa=4*rand()-2 # between -2 and 2
xb=4*rand()-2 # between -2 and 2
yb=f(xb)-2*rand()
ya=f(xa)+2*rand()
```

- Try to find an example where the optimal path is not the union of two segments.
- In the cases where it is the union of two segments, define a cost function that relates the position on the graph to the time of travel between A and B.
- Use the above programs (dichotomic search with centered difference method) to locate the optimal path with $v_1 = 1$ and $v_2 = 2$.
- Draw a nice graphical solution.

```
X=linspace(-2,2)
plot(X,f(X))
fill_between(X,f(X),0,alpha=0.3)
plot([xa,xm,xb],[ya,f(xm),yb],'-ro')
text(xa,ya+.1,"$A$",color='r',
                horizontalalignment='center')
text(xb,yb-.2,"$B$",color='r',
                horizontalalignment='center')
text(xm+.1,f(xm)+.2,"$M_*$",color='r',
                horizontalalignment='center')
text(0,.2,"$Z_2$ : vitesse $v_2$",color='b',
                horizontalalignment='center')
text(0,3,"$Z_1$ : vitesse $v_1$",
                horizontalalignment='center')
axis('scaled')
show()
```

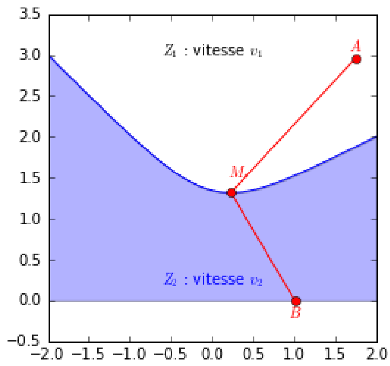

FIGURE 4 – Shortest path.