EPFL Fall 2018
Doctoral School
Civil and Environmental Engineering

Scientific Programming
for Engineers

Week 4 — *Homework: Conjugate Gradient*

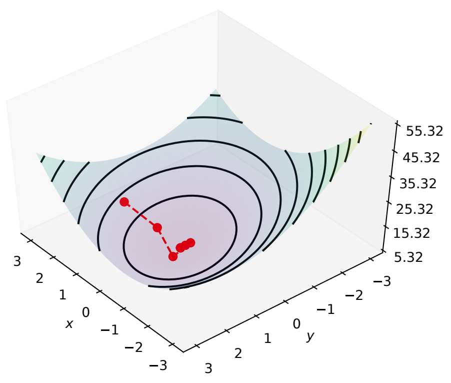
The goal of this exercise is to get familiar with scipy and numpy module by implementing an iterative solver and visualizing the results. We will use Python for this exercise.

Exercise 1: *Scipy optimization*

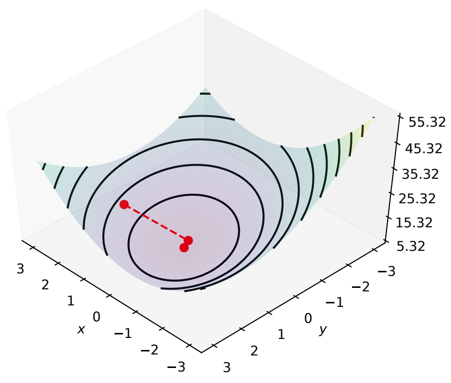
In this exercise, we will use the optimization library of **Scipy** module. The goal of this exercise is to find the value of (x, y) which minimizes the function $S(x, y)$ defined by the equation:

$$S(x, y) = 2x^2 + \frac{3}{2}y^2 + xy - x - 2y + 6$$

Figure below shows the surface plot for $S(x, y)$. The red line shows the iterative path taken by two different solvers (BFGS, Conjugate gradient) to locate the minima of the function.



BFGS



Conjugate Gradient

1. Create a **optimizer.py** file. In this file, create a function which takes $S(x, y)$ as a input argument in form of a python functor and returns the minimizer. Use `scipy.optimize.minimize` routine to solve the minimization problem. The type of solver can be specified through `method` parameter. For more help, refer to the documentation : Scipy optimize

2. Implement an another routine which plots the $S(x, y)$ and the solution at each iteration step similar to figure shown above. Use **Matplotlib** module for plotting. To get the solution at each iterative step from `scipy.optimize.minimize`, you will have to use the parameter `callback`. The argument takes a functor. The functor definition is as given:

```python
def getIterationSteps(x):
# x contains the solution at each step
# use x as you want
```

Exercise 2: *Conjugate Gradient*

The conjugate gradient (CG) method is one of the many known iterative techniques for solving sparse symmetric definite systems of linear equations. For this exercise, implement the CG method to find the minimizer of the quadratic function

$$\phi(\underline{x}) = \frac{1}{2}\underline{x}^T \underline{\underline{A}}\ \underline{x} - \underline{x}^T \underline{b}$$

The minimizer of the above function is also the solution of the linear system of equation

$$\underline{\underline{A}}\ \underline{x} = \underline{b}$$

1. Create a **conjugate_gradient.py** file which implements CG method. Use `einsum` module **Numpy** to perform different matrix and vector operations such as inner product, dot product etc. The implementation should be independent of matrix and vector size. For help, please refer to the wikipedia page Conjugate Gradient
2. Solve for the function $S(x, y)$ given in the previous exercise.
3. Use the plotting routine developed in previous exercise to plot the solution at each iteration step and compare the method with the **Scipy** optimization method chosen by you in the previous exercise.