Lab 1 (minimization of a quadratic function)

1) Write a Matlab script test1.m able to find the minimum $\underline{a}_{opt}$ of the function

$$f(\underline{a}) = \left\| \underline{\underline{X}}\,\underline{a} - \underline{c} \right\|^2$$

where

$$\underline{\underline{X}} = \begin{bmatrix} 1 & 0.2 \\ 0.1 & 1 \end{bmatrix}, \quad \underline{c} = \begin{bmatrix} 0.1 \\ 0.5 \end{bmatrix}$$

(the correct solution is $\underline{a}_{opt} = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$).

Use the gradient algorithm, contained in a Matlab function gradient_test1.m

[a_opt1] = gradient_test1(X,C,eps,gamma,M)

where **eps** is the stopping parameter , so that the subroutine will stop if

```
norm(a(:,index+1)-a(:,index))<eps
```

gamma is the step of the gradient algorithm, and **M** is the maximum number of steps performed by the gradient algorithm. Notice that large values of gamma will make the algorithm move faster towards the solution, but may make your solution unstable, while small values of gamma will make the algorithm move slowly but more surely towards a minimum. Furthermore, small values of eps can insure that you find the minimum value with high precision, while large values of eps may not guarantee that you are exactly at the minimum value. Finally, notice that the algorithm finds one minimum, but if several minima are present, it may find a local minimum.

2) Set initially eps=1 and gamma=1, and see what happens. How many steps did the algorithm perform? Decrease the values of eps and gamma (for instance eps=0.1 and gamma=0.1) and describe what happens. Write your findings inside the program as comments (using %). You should increase or decrease eps and gamma? Which parameter should be smaller?

3) Adjust your parameters until you find the correct minimum and write the reason of your choice.
4) Modify the stopping criterion of the gradient algorithm as

$$\left| f\left(\underline{a_{i+1}}\right) - f\left(\underline{a_{i+1}}\right) \right| < \epsilon$$

and say if things improve or not. In Matlab the condition can be written as

```
norm(norm(X*a(:,index+1)-C).^2-norm(X*a(:,index)-C).^2)<eps
```

5) Use the gradient algorithm with **adaptable step-size gammai** (steepest descent) and say it things improve. In order to do this, you can write a second Matlab function

gradient_test2.m [a_opt2] = gradient_test2(X,C,eps,M)

where the parameter gammai will be recalculated ad every step inside the function as

$$\gamma_i = \frac{\|\nabla f(\underline{a}_i)\|^2}{\nabla f(\underline{a}_i)^T \underline{\underline{\mathrm{H}}}(\underline{a}_i) \nabla f(\underline{a}_i)}$$, where $\underline{\underline{\mathrm{H}}}(\underline{a}_i) = 2\underline{\underline{X}}^T\underline{\underline{X}}$ and $\nabla f(\underline{a}_i) = 2\underline{\underline{X}}^T\underline{\underline{X}}\ \underline{a}_i - 2\underline{\underline{X}}^T\underline{c}$

In Matlab these equations can be written as

```
grad=2*((X')*X)*a(:,index)-2*X'*C;
gammai=norm(grad).^2/(2*grad'*(X'*X)*grad);
```

6) Write your answers as comments in your program. Publish your final program and results using the option publish (pdf output) and upload it on the course website.
7) You may find the programs test1.m and gradient_test1.m written in class on the course website. They also contain the commands required to plot the function $f(\underline{a})$ in 3D.