

Numerical Optimization - Project 1, First Phase

The overall task is to program the four methods for unconstrained optimization: steepest descent, Newton method, conjugate gradient method and quasi-Newton method.

It is up to you whether you choose the line search or the trust region approach.

It is also up to you which of the (nonlinear) conjugate gradient methods and which of the quasi-Newton methods you do.

In the first phase, you should just program these methods and test each of them on

- 5 problems with 10-20 variables for which you know the solution - you can use just a quadratic objective function;
- 5 problems with 1 or 2 variables for which you know the solution - here, do not use a quadratic objective, you can use e.g. polynomials of the degree higher than 2, but also other types of functions.

For instance, you can construct a problem of the first type as follows. First, generate a positive definite n by n matrix A for $10 \leq n \leq 20$. Then, choose a vector $\bar{x} \in \mathbb{R}^n$ and set $b = A\bar{x}$. Then \bar{x} is the solution of the quadratic problem $\min f(x) = 1/2x^T Ax - bx$, since $\nabla f(\bar{x}) = A\bar{x} - b = 0$.

Constructing a polynomial with a known minimizer is not very difficult. For instance, $(x-a)(x-b)(x-c)$ is a polynomial of degree 3 with roots at a , b and c . Thus, one can integrate it to obtain a polynomial of degree 4 with stationary points at a , b and c .

After you test your programs on these 10 problems, please, send me/upload the codes together with the tests and the information how many problems out of 10 you successfully solved, e.g. 8/10 or 9/10. A problem is successfully solved if all 4 methods find a minimizer (from a suitable starting point).

It should not be difficult to program the methods well enough so that they work on simple problems. If you fail to solve a problem, you should try to find an issue in the code and improve it. Anyway, it is up to you which 10 problems you solve, so if you encounter a problem you can not solve, you can replace it.

Then, in the second phase, you will work in teams and you will try to code the methods more carefully (details later). Your individual final score will be computed as the score of your team multiplied by the fraction of problems you solved (e.g. 8/10 or 9/10).