Numerical Optimization 2024 - Homework 6

Deadline: Wednesday, May 8, 15:30.

In this homework, you will apply various quasi-Newton methods to minimize

$$f(x) = \frac{1}{2}x^T Q x - b^T x$$

where $b = (1,1,\ldots,1) \in \mathbb{R}^n$ and Q is the Hilbert matrix of dimension n given by $q_{ij} = 1/(i+j-1)$ for $i,j = 1,\ldots,n$ (q_{ij} denotes the element of Q in row i and column j). For the starting point, always choose $x_0 = (0,0,\ldots,0) \in \mathbb{R}^n$ and for the starting matrix $B_0 = I$ (identity matrix with 1 on the diagonal and 0 elsewhere). The idea is to get more familiar with the method by trying to do some steps (Problems 1 and 2 should be computed by hand(!) ($n = 2$) and Problem 3 by program ($n = 5$)) and to see the manifestation of the theory from Sections 6.2 and 6.3 from the Nocedal-Wright book. The homework should not be difficult, but it requires you to do some computations.

Problem 1. (Testing Theorem 6.1 - SR1 method)

For dimension $n = 2$, perform the SR1 iteration: since Theorem 6.1 is not effected by step size, use simply the update (6.28) together with the SR1 update (6.24) for B_k and (6.25) for the inverse matrix $H_k := B_k^{-1}$. At each iteration, compute not only the iterates but also both matrices B_k and H_k as well as number $(s_k - H_k y_k)^T y_k$. Check if the statement of Theorem 6.1 holds. Stop the iteration when you reach the solution - think about know how many iterations you will need to find the solution.

Problem 2. (Testing Theorem 6.3 - DFP method)

For dimension $n = 2$, perform the DFP iteration: even though Theorem 6.3 is also not effected by step size, **do not** use the update (6.34); use the exact step length (α_k is given by formula (3.25)) together with the DFP update (6.13) for B_k and (6.15) for the inverse matrix $H_k := B_k^{-1}$. At each iteration, compute not only the iterates but also both matrices B_k and H_k together with the matrix in (6.35), eigenvalues of this matrix, and quantities in (6.36). Check if the statement of Theorem 6.3 holds. Stop the iteration when you reach the solution - again think about know how many iterations you will need to find the solution (see also Theorem 6.4).

Problem 3.

Program the SR1 and the DFP update (as used in Problems 1 and 2) and do exactly the same tasks as in Problems 1 and 2 but for dimension $n = 5$. Moreover, do the tasks of Problem 2 also for the SR1 methods (for $n = 2$ and $n = 5$) - do you expect the statement of Theorem 6.3 to be valid? Is it?