2102 509 – Introduction to Optimization Techniques
Homework # 3: Due date Monday 3th November 2025 by 11:00 am.
Submit your homework (hard copy) in the classroom.
Late homework will not be accepted. Duplication of homework is strictly forbidden.

---

1. (10 points) Let $Q \in \mathbb{R}^{n\times n}$ be a positive definite symmetric matrix. Given an arbitrary set of linearly independent vectors $\left\{p^{(0)}, \ldots, p^{(n-1)}\right\}$ in $\mathbb{R}^n$, the *Gram-Schmidt* procedure generates a set of vectors $\left\{d^{(0)}, \ldots, d^{(n-1)}\right\}$ as follows:

$$\begin{aligned} d^{(0)} &= p^{(0)} \\ d^{(k+1)} &= p^{(k+1)} - \sum_{i=0}^{k} \frac{p^{(k+1)T} Q d^{(i)}}{d^{(i)T} Q d^{(i)}} d^{(i)}, \quad k = 0, 1, 2, \ldots \end{aligned}$$

Show that the vectors $d^{(0)}, \ldots, d^{(n-1)}$ are Q-conjugate.

2. (10 points) Let Q be an $n \times n$ real symmetric matrix.

(a) Show that there exists a Q-conjugate set $\left\{d^{(1)}, \ldots, d^{(n)}\right\}$ such that each $d^{(i)}$ ($i = 1, 2, \ldots, n$) is an eigenvector of Q.

(b) Suppose that Q is positive definite. Show that if $\left\{d^{(1)}, \ldots, d^{(n)}\right\}$ is a Q-conjugate set that is also orthogonal (*i.e.*, $d^{(i)T} d^{(j)} = 0$ for all $i, j = 1, 2, \ldots, n$ where $i \neq j$) and if $d^{(i)} \neq 0$, $i = 1, 2, \ldots, n$, then each $d^{(i)}$ ($i = 1, 2, \ldots, n$) is an eigenvector of Q.

3 (10 points) Let x_k and s_k be, respectviely, an iterate and a search vector with descent property. Develop a line-search procedure for computing a step-length λ_k that satisfies the following criteria:

$$\begin{aligned} f(x_k + \lambda_k s_k) &\leq f(x_k) + \mu \lambda_k s_k^T \nabla f(x_k) && \text{(Armijo's condition)} \\ \left| s_k^T \nabla f(x_k + \lambda_k s_k) \right| &\leq -\eta s_k^T \nabla f(x_k) && \text{(Strong Wolf's condition)} \end{aligned}$$

where $0 < \mu < \eta < 1$, and $\lambda_k = 1$ is used whenever it satisfies the above criteria. Observe that as η is close to zero, the corresponding line search becomes exact.

First explain the concepts you use for each step. Then write the procedure in the same form as the algorithms in the handouts.

4. (30 points) Write your MATLAB function for solving the n-dimensional minimization problem using the BFGS method with the line-search procedure that is developed above.

The format of the function BFGS is as follows.

function [x_{min},f_{min},Xk,Fk,Gk,Lk,nF,nG,IFLAG] = BFGS(FcnName,x_0,epsilon,mu,eta,itmax)

where x_{min} is the obtained estimate of the minimizer of f; f_{min} is the value of $f(x_{min})$; Xk is the array containing iterates x_k ($k = 0, 1, 2, \ldots$); Fk, Gk and Lk are the arrays containing $f(x_k)$, $\nabla f(x_k)$ and λ_k ($k = 0, 1, 2, \ldots$) respectively; nF and nG are the arrays of the numbers of evaluation of the objective function f and the gradient ∇f, respectively, in each iteration; IFLAG is is a flag variable that indicates the status of return from calling BFGS; x_0 is the given starting point; epsilon is the parameter used in the stopping criterion for the method; mu and eta are parameters in the line search criteria (if eta is chosen to be very small, the line search process becomes exact); itmax denotes the maximum number of the iterations allowed.

Print out your program. Write comments in your codes so that it is readable and can be graded. Then use the above program to determine the minimum of the Rosenbrock's function for 2 cases: eta $= 0.1$ (fairly accurate line search) and eta $= 0.98$ (inaccurate line search) while mu is set to be 10^{-4} for both cases. Locate the value of x^* with at least 4 significant digits. Explain why you choose all the parameters used in solving the problem. Tabulate – **for each iteration** – the following results: x_k, $f(x_k)$, the number of function evaluations, the number of gradient evaluations. Compare the results from the two cases and make a discussion. You may run more cases if you think this can support your discussion.

The MATLAB function sees Rosenbrock's function in the following format:

```
function [f, gradient] = Rosenbrock( x, options ).
```

If options = 1, only f is computed. If options = 2, f and gradient are computed.

Suchin Arunsawatwong, 27th October 2025.