Lab2 : Unconstrained Optimization

Exercise 1.
Quadratic optimization problems
Consider

$$\min J(x) := \frac{1}{2}x'Ax - b'x + c \tag{P}$$

where A is an $n \times n$ matrix, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$

1. When A is symmetric, prove that $\nabla J(x) = Ax - b$.
2. Prove that (P) has no optimal solution when A is symmetric and admits at least one negative eigenvalue.
3. Prove that (P) has a unique solution when A is symmetric positive definite.

Exercise 2.
Gradient type algorithms
We consider the 3 following algorithms.

G.M.C : Gradient Method with Constant step-size
- Choose $x^0 \in \mathbb{R}^n$ and $\rho > 0$.
- for $k = 0; 1; \dots$:
  - Compute $d_k = -\nabla J(x^k)$,
  - Define $x^{k+1} = x^k + \rho d_k$
- Stop if $\|\nabla J(x^{k+1}\| \leq prec$

G.M.O : Gradient Method with Optimal step-size (Quadratic optimization problems)
- Choose $x^0 \in \mathbb{R}^n$.
- for $k = 0; 1; \dots$:
  - Compute $d_k = -\nabla J(x^k)$,
  - Compute $\rho_k := \frac{d'_k d_k}{d'_k A d_k}$
    (explain this choice of step-size)
  - Define $x^{k+1} := x^k + \rho_k d_k$
- Stop if $\|\nabla J(x^{k+1}\| \leq prec$

C.G.M : Conjugate Gradient Method (Quadratic optimization problems)
- Choose $x^0 \in \mathbb{R}^n$, let $r_0 := \nabla J(x^0)$ and $d_0 := -r_0$.
- for $k = 1; 2; \dots$:
  - $\rho_k := \frac{r'_k r_k}{d'_k A d_k}$, Define $x^{k+1} := x^k + \rho_k d_k$
  - $r_{k+1} := \nabla J(x^{k+1})$, $\beta_k := \frac{r'_{k+1} r_{k+1}}{r'_k r_k}$
  - $d_{k+1} := -r_{k+1} + \beta_k d_k$.
- Stop if $r_{k+1} \leq prec$

1. Write 3 matlab functions : GMC.m, GMO.m and CGM.m and test them on randomly generated quadratic problems.
2. Write a matlab function construc.m "$[A, b] = construct(n)$" to get for $n \in \mathbb{N}$ a vector $b = (1, \dots, 1)^t \in \mathbb{R}^n$ and an $n \times n$ matrix

$$A = \begin{pmatrix} 4 & -2 & 0 & \dots & -1 \\ -2 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -2 \\ -1 & \dots & 0 & -2 & 4 \end{pmatrix}$$

3. Prove (using matlab) that such A is positive definite.
4. Test the 3 algorithms for a maximum number of iterations (Maxiter=100), when $n = 10$, $prec = 10^{-10}$ (and $\rho = 0.1$ for G.M.C).

5. We want to check numerically that the number of iterations to reach a given precision depends on the condition number of A.
   Theoretically, the number of iterations is proportional to $cond_2(A)$ for G.M.C and G.M.O. For C.G.M it is proportional to $\sqrt{cond_2(A)}$.
   Test from an array of dimesions $arrn = 10:30$ and give the corresponding graphs
   $cond_2(A) = f(item)$.
6. Let now A $(n \times n)$ be a tridiagonal symmetric matrix with $A_{i,i} = 3i^2$, $i = 1,...,n$ and $A_{i,i+1} = 1$, $i = 1,...,n-1$.
   — Write a new matlab function to construct such matrix A for any value of n.
   — Compute $cond_2(A)$ for $n = 10,\ 100,\ 1000$. Comment
   — For $n = 1000$, $prec = 10^{-10}$ and $itermax = 2000$, test C.G.M to solve this new problem.
   — Consider the $n \times n$ diagonal matrix C defined by $C_{i,i} = \frac{1}{i}$ and define $\tilde{A} := CAC$ and $\tilde{b} = Cb$.
   Compute $cond_2(\tilde{A})$ for $n = 10,\ 100,\ 1000$. Comment.
   — Show that $\tilde{A}\tilde{u} = \tilde{b} \Leftrightarrow Au = b$, and $u = C\tilde{u}$.
   — For $n = 1000$, $prec = 10^{-10}$ and $itermax = 2000$, test C.G.M to solve the new problem (with $\tilde{A}$ and $\tilde{b}$). Comment.

Exercise 3.

Extensions to unconstrained nonlinear problems

To solve nonlinear non quadratic optimization problems, we can do the following adaptations . Let $f : \mathbb{R}^n \to \mathbb{R}$ and $g(x) = \nabla f(x)$

S.D.M : Steepest Descent Method

— Replace in G.M.O, the definition of ρ_k by

$$\rho_k = \mathrm{argmin}\{\mathrm{f}(\mathrm{x^k} + \rho \mathrm{d_k} \mid \rho \geq 0\}$$

or compute ρ_k using Armijo.

F.R : Fletcher Reeves Method

— In C.G.M, replace the definition of ρ_k by

$$\rho_k = \mathrm{argmin}\{\mathrm{f}(\mathrm{x^k} + \rho \mathrm{d_k} \mid \rho \geq 0\}$$

or compute ρ_k using Armijo.

— Change the definition of β_k : to $\beta_k := \frac{\|g(x^{k+1}\|^2}{\|g(x^k\|^2}$.

1. Write 2 matlab functions corresponding to S.D.M and F.R.
2. Test these functions for
   (a) $f(x) = x_1^2 + 6x_1x_2 + 25x_2^2 - 12x_1 - 2x_2 - 6$

   (b) Rosenbrock

   $$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

   (c) Himmelblau

   $$f(x_1, x_2) = (x_2 + x_1^2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

   (d) DENNI $\quad F(x) = \displaystyle\sum_{1 \leq k \leq n} k.x_k^2 + \left(\sum_{1 \leq k \leq n} x_k \right)^2.$

   (e) VAR $\quad F(x) = \displaystyle\sum_{1 \leq k \leq n} x_k^2 + \left(\sum_{1 \leq k \leq n} \sqrt{k} x_k \right)^2 + \left(\sum_{1 \leq k \leq n} \sqrt{k} x_k \right)^4.$

   (f) MONDIA $\quad F(x) = \displaystyle\sum_{2 \leq k \leq n} 100(x_1 - x_k^2)^2 + \sum_{1 \leq k \leq n} (1 - x_k)^2 .$

Exercise 4.

Second order Newton type methods

Classical Newton Method

— choose an initial point x^0 near a solution

— for $k \geq 0$,
  — compute $\nabla f(x^k)$ et $\nabla^2 f(x^k)$.
  — solve the linear system
    $\boxed{\nabla^2 f(x^k)\, d_k = -\nabla f(x^k)}$
  — define $x^{k+1} := x^k + d_k$ (use in a second code an Armijo like stepsize)
  — Stop if $\|\nabla f(x^{k+1}\| \leq prec$

The well known BFGS Method

— choose an initial point x^0 and a symmetric positive definite matrix H_0 (default one can be the identity matrix). Compute $g_0 := \nabla f(x^0)$.

— for $k \geq 0$,
  — compute $d_k = -H_k g_k$.
  — compute α_k minimising $\phi(\alpha) := f(x^k + \alpha d_k)$ (or use armijo)
  — define $x^{k+1} = x^k + \alpha_k d_k$, then let $p_k := \alpha_k d_k$, $g_{k+1} = \nabla f(x^{k+1})$ and $q_k := g_{k+1} - g_k$.
  — Stop if $\|\nabla f(x^{k+1}\| \leq prec$
  — update the matrix H_{k+1} using the following (CR_2) formula.

$$\boxed{(CR_2) \quad H_{k+1} = H_k + \left(1 + \frac{q_k^T H_k q_k}{q_k^T p_k}\right) \frac{p_k p_k^T}{p_k^T q_k} - \frac{p_k q_k^T H_k + H_k q_k p_k^T}{q_k^T p_k}}$$

1. Write 2 matlab functions corresponding to Newton and BFGS methods.
2. Test these functions using the examples in exercise 3. Comment.