Université de Versailles — Année 2024-2025
`http://dumas.perso.math.cnrs.fr/M1-optim.html` — semestre 1

TP1: méthodes à région de confiance, problème sans contraintes

L'objectif de cette séance est d'utiliser le logiciel Python pour implémenter une méthode de type région de confiance pour la minimisation locale d'une fonction f définie sur $\mathbb{R}^n$. Celle-ci s'écrit, pour un point de départ X_0 et une région de confiance de rayon initial Δ_0, sous la forme pseudo-algorithmique suivante (issu de l'article Gould -Leyffer disponible sur le site):

> Given $k = 0$, $\Delta_0 > 0$ and x_0, until "convergence" do:
> Build the second-order model $m(s)$ of $f(x_k + s)$.
> "Solve" the trust-region subproblem to find s_k
> for which $m(s_k)$ "<" f_k and $\|s_k\| \leq \Delta_k$, and define
>
> $$\rho_k = \frac{f_k - f(x_k + s_k)}{f_k - m_k(s_k)}.$$
>
> If $\rho_k \geq \eta_v$ [*very successful*] $\boxed{0 < \eta_v < 1}$
> set $x_{k+1} = x_k + s_k$ and $\Delta_{k+1} = \gamma_i \Delta_k$. $\boxed{\gamma_i \geq 1}$
> Otherwise if $\rho_k \geq \eta_s$ then [*successful*] $\boxed{0 < \eta_s \leq \eta_v < 1}$
> set $x_{k+1} = x_k + s_k$ and $\Delta_{k+1} = \Delta_k$.
> Otherwise [*unsuccessful*]
> set $x_{k+1} = x_k$ and $\Delta_{k+1} = \gamma_d \Delta_k$. $\boxed{0 < \gamma_d < 1}$
> Increase k by 1.

Exercice 1

On cherche à programmer la méthode de type région de confiance, avec un modèle quadratique d'approximation d'ordre 2. Dans un premier temps, on utilisera une librairie d'optimisation de Python pour résoudre la minimisation du modèle quadratique dans la région de confiance (par exemple la fonction `lsq_linear` de la librairie `scipy.optimize`).

1. Ecrire une fonction Python ayant pour arguments J, ∇J, HJ, η_v, γ_i, η_s, γ_d (paramètres de la région de confiance), X_0 (point initial) et N (nombre d'itérations) et renvoyant la valeur de X_N .

2. On cherche à minimiser la fonction suivante:

   $$J(x, y) = (y - 2)^4 + (x - 1)^4$$

   On prend pour cela $(\eta_v, \gamma_i, \eta_s, \gamma_d) = (0.9, 2, 0.1, 0.5)$ et $X_0 = (1, 0)$.

   Vérifier la bonne convergence de la méthode.

   On pourra ensuite tester la méthode sur le cas de la fonction de Rosenbrock.

Exercice 2

On s'intéresse à présent à la résolution du problème quadratique sur la région de confiance. On admet le résultat suivant: tout minimum global s^* de la fonction

$$m(x+s) = < g, s > + \frac{1}{2} < Bs, s >$$

sur $B(0, \Delta)$ est tel que

$$(B + \lambda_* I)s^* = -g$$

où $B + \lambda_* I$ est semi-définie positive, $\lambda_* \geq 0$ et $\lambda_*(||s^*|| - \Delta) = 0$. Si de plus $B + \lambda_* I$ est définie positive, alors ce minimum est unique.

On peut alors montrer que l'algorithme suivant basé sur une succession de décomposition de Cholesky d'un matrice définie positive:

Let $\lambda > -\lambda_1$ and $\Delta > 0$ be given.
Until "convergence" do:
Factorize $B + \lambda I = LL^T$.
Solve $LL^T s = -g$.
Solve $Lw = s$.
Replace λ by

$$\lambda + \left(\frac{\|s\|_2 - \Delta}{\Delta} \right) \left(\frac{\|s\|_2^2}{\|w\|_2^2} \right).$$

est convergent si $\lambda \in [-\lambda_1, \lambda_*]$ où λ_1 désigné la plus petite valeur propre de B.

1. Programmer avec Python l'algorithme précédent de recherche du minimum global d'une fonction m de type précédent sur $B(0, \Delta)$ (arguments: B, g et Δ).

2. Tester la méthode de région de confiance avec ce nouvel algorithme.