Nom :
Prénom :
Numéro de groupe (1 à 6) :

Calcul Scientifique - TP2

Sous Linux, ouvrez un terminal ("Applications" $\rightarrow$ "Outils système" $\rightarrow$ "Terminal") et copiez-collez les instructions du script `http` ://cermics.enpc.fr/cours/CS/launch_TP2.txt. Cela téléchargera les sources du code, créera le répertoire d'accès et ouvrira les applications nécessaires. Des informations supplémentaires sont ensuite données dans le script. Vous pourrez trouver une aide pour Scilab en tapant "man scilab" sous google par exemple.

Attention ! Les scripts que vous récupérerez ainsi ne sont pas prêts à l'usage. Il faut au préalable compléter les lignes où apparaissent ♯♯

On considère la fonction

$$V(x,y) = (x^3 + y^2)\mathrm{e}^{-(x^2+y^2)}$$

que l'on cherche à minimiser. Dans la suite, on considérera le problème de minimisation sans contrainte, et le problème de minimisation sous la contrainte $\Phi(x,y) = (x-a)^2+(y-b)^2-\rho^2 = 0$. On peut observer qualitativement les variations de V avec les options graphiques `Contour = 0` puis `Contour = 1` en faisant tourner les figures (sans lancer les routines d'optimisation, *i.e.* en conservant `algo = 0`). Pour faire tourner les figures, on utilisera la fonction `3DRot`. Avec les options `Contour = 2` et `indication_valeur = 1`, on peut localiser plus précisément les positions des extremas locaux et des points selles, ainsi que les valeurs approximatives de V en ces points :

- Le minimum absolu est autour de $(-1.22, 0)$ et V vaut environ -0.39 en ce point. Le maximum absolu autour de $(1.20, 0)$, sa valeur est 0.39. On a des maxima locaux en $(0, \pm 1)$, de valeurs 0.36. Il y a également des point-selles en $(0, 0)$ (valeur 0) et $(0.65, \pm 0.85)$ (valeur 0.32).
- Lorsque la contrainte est présente pour les valeurs $(a, b) = (0.9, 0.6)$ et $\rho = 1$, on a un minimum global en (0.1,0), avec une valeur autour de 0 ; un minimum local en (1.6,1.35), avec une valeur autour de 0.9 ; un maximum global en (1.2,-0.4), avec une valeur autour de 0.4 ; un maximum local en (0,1), avec une valeur autour de 0.36 – mais pas de point selle vu que cela revient à des minimisations en dimension 1 !

Question 1 : Implémentation de la méthode du gradient. Calculer le gradient (remplacer les champs ♯♯ dans la fonction `nablaV` par les expressions des dérivées partielles).

Réponse :

Question 2 : Minimisation sans contrainte. On va à présent procéder à la minimisation de V par un algorithme de gradient à pas fixe, en partant de la condition initiale $(x^0, y^0) = (-1, 1.5)$. On prendra `algo = 1`, `contrainte = 0`, `N_contour = 30` et `indication_valeur = 0`. Etudier la convergence de l'énergie en fonction du pas λ de l'algorithme de gradient : y a-t-il convergence ou non, la convergence de l'erreur est-elle exponentiellement rapide, et pour quelle valeur de λ cette convergence est la plus rapide ? On fixera dans le code les valeurs de `lambda`, de la précision `eps`, et du nombre maximal d'itérations `max_iter` (signalées par $\sharp\sharp$).

Réponse :

Question 3 : Minimisation avec contrainte. Reprendre la question précédente pour la minimisation sous contrainte (option `contrainte = 1`) avec les valeurs $(a, b) = (0.9, 0.6)$ et $\rho = 1$. On utilisera un algorithme de gradient à pas fixe projeté, pour lequel on commencera par implémenter la projection sur $\Phi^{-1}\{0\}$ (fonction `projection`). On discutera également la condition d'Euler (le vecteur dessiné en noir est proportionnel à ∇V et celui en rouge à $\nabla \Phi$).

Réponse :

Questions subsidiaires.
Reprendre les questions précédentes avec des conditions initiales aléatoirement choisies (option `CI = 2`).
Etudier l'influence du critère d'arrêt (variations de l'énergie plutôt que norme euclidienne de la différence entre les itérés).