Méthodes numériques : optimisation.
Examen du 13 mai 2016

Recommandations

L'examen est probablement long, mais le barème en tiendra compte. Les trois exercices sont indépendants. Pour les questions demandant d'écrire du code en python (ou pour celles dont des morceaux de code sont déjà donnés), on supposera que les librairies `numpy` et `matplotlib` ont déjà été chargées, par exemple dans IPython avec la directive `%pylab inline`.

Les questions de chaque exercice ne sont pas censées être bloquantes pour les suivantes. N'hésitez donc pas à admettre des résultats et passer à la suite.

1 Préconditionnement de la méthode du gradient conjugué

Le contexte de cet exercice est l'étude numériques de (grands) graphes aléatoires. On se donne N éléments (par exemple des personnes d'un réseau social) numérotés de 0 à $N-1$, et on suppose que l'on a relié certains points entre eux (par exemples si les personnes sont amies), c'est à dire que l'on a une liste d'arêtes (i,j) reliant les points numérotés i et j (avec $i \neq j$).

Voici comment on génère une telle liste d'arêtes en python :

```
1  N=10**4
2  listearetes=[]
3  for k in range(10*N):
4      i,j=(randint(N),randint(N))
5      if i!=j:
6          listearetes.append((i,j))
```

1. On se donne un réel δ, représenté par la variable `delta` ci-dessous, et on construit un vecteur d dans $\mathbb{R}^N$ (correspondant à la variable `diag` ci-dessous), puis une fonction $\mathcal{A} : \mathbb{R}^N \to \mathbb{R}^N$ (correspondant à la définition de fonction `fA`) que l'on code comme suit :

```
7  delta=2.5
8  diag=delta*ones(N)
9  for (i,j) in listearetes:
10     diag[i]+=1
11     diag[j]+=1
```

```
12 def fA(x):
13     global compteurfA
14     compteurfA+=1
15     fAx=x*diag
16     for (i,j) in listearetes:
17         fAx[j]-=x[i]
18         fAx[i]-=x[j]
19     return fAx
```

Montrer que pour $x \in \mathbb{R}^N$, ce code correspond à avoir $\mathcal{A}(x) = Ax$, où $A = (a_{i,j})_{0 \leqslant i,j < N}$ est une matrice symétrique de $M_N(\mathbb{R})$, et expliciter ses coefficients a_{ij}. On pourra distinguer les cas où $i = j$ et $i \neq j$, et on rappelle qu'en python, le symbole $*$ correspond à la multiplication élément par élément (de deux vecteurs dans les cas ci-dessus).

On suppose pour la suite (sauf pour la question 6) que la matrice A est définie positive.

2. On cherche à approcher numériquement la solution de $Ax + b = 0$, où $b \in \mathbb{R}^N$ est donné. Pourquoi ne stocke-t-on pas A sous la forme d'une matrice ? Quelles méthodes peut-on alors utiliser ?

3. On se donne une matrice symétrique inversible M, que l'on suppose écrite sous la forme EE^T.
   (a) Montrer que résoudre $Ax + b = 0$ équivaut à résoudre $E^T x = \hat{x}$, où $\hat{x}$ est un point critique de la fonction $\hat{f} : \hat{x} \mapsto \frac{1}{2}\langle \hat{x}, \hat{A}\hat{x}\rangle + \langle \hat{b}, \hat{x}\rangle$, où $\hat{A} = E^{-1}A(E^{-1})^T$ et $\hat{b} = E^{-1}b$.
   (b) Écrire les relations de récurrence (et leur initialisation) satisfaites par les itérées $\hat{x}_k$, $\hat{r}_k$ et $\hat{p}_k$ (points, gradients, et directions de descente) de la méthode du gradient conjugué pour la minimisation de la fonction $\hat{f}$ (on suppose que $\hat{A}$ est définie positive), partant du point $\hat{x_0} = 0$. On demande d'écrire la version des relations de récurrence correspondant à ce qui a été appelé « présentation standard de l'algorithme » dans le cours.
   (c) En déduire que si l'on a $E^T x_k = \hat{x}_k$, $E^{-1} r_k = \hat{r}_k$ et $E^T p_k = \hat{p}_k$, alors on obtient pour tout $k \geqslant 0$, dès que $r_k \neq 0$:

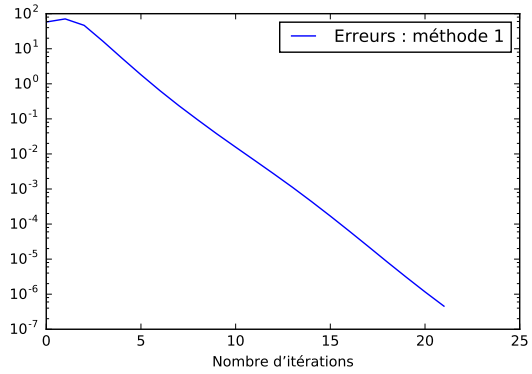
$$\begin{aligned} x_{k+1} &= x_k + \alpha_k p_k, \\ r_{k+1} &= r_k + \alpha_k A p_k, \\ p_{k+1} &= -M^{-1} r_{k+1} + \frac{\langle r_{k+1}, M^{-1} r_{k+1}\rangle}{\langle r_k, M^{-1} r_k\rangle} p_k, \end{aligned}$$

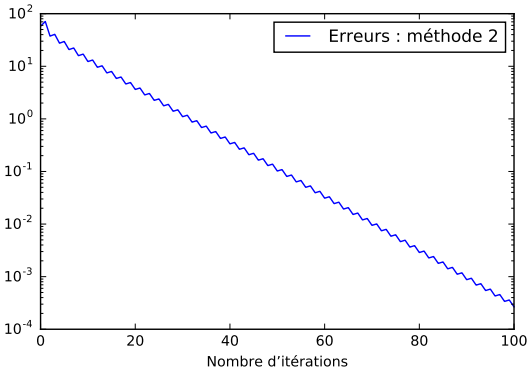
   avec $\alpha_k = \frac{\langle r_k, M^{-1} r_k\rangle}{\langle p_k, A p_k\rangle}$. Que valent x_0, r_0 et p_0 ? En déduire que l'on peut calculer les x_k, r_k et p_k sans avoir connaissance de la décomposition de M en EE^T.
4. On veut programmer cette méthode. On se donne les deux fonctions suivantes, pour calculer le produit scalaire dans $\mathbb{R}^N$ et l'inverse de M.

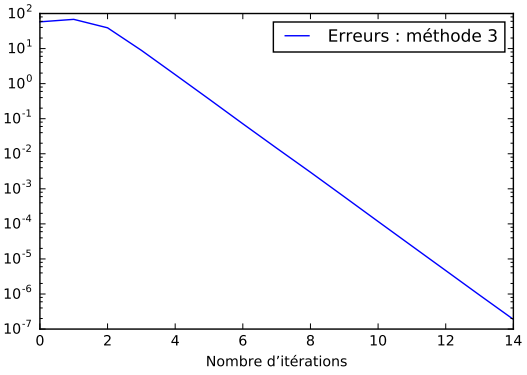
```
20 def prodscal(x,y):
21     global compteurps
22     compteurps+=1
23     return sum(x*y)
```

```
24 invdiag=1/diag
25 def invM(x):
26     global compteurinvM
27     compteurinvM+=1
28     return invdiag*x
```

   (a) Que vaut la matrice M dans ce cas ? Pourquoi n'a-t-on pas stocké M^{-1} directement sous la forme d'une matrice ?
   (b) Écrire en python l'algorithme de la question précédente, sous la forme d'une fonction qui prend en argument b, une tolérance relative ε, et un nombre maximal d'itérations (que l'on peut fixer à 100 par défaut), et qui renvoie la solution approchée x_K et la liste des $\|r_k\|^2$ pour $0 \leqslant k \leqslant K$, où K est le nombre d'étapes effectuées pour atteindre la tolérance donnée (ou le nombre maximal d'itérations s'il est atteint). On s'attachera à faire appel le moins de fois possible par étape aux fonctions permettant de calculer $\mathcal{A}(x)$, $M^{-1}x$ et le produit scalaire.
   (c) Écrire en python les instructions correspondant :
   — au test de la méthode pour un vecteur b aléatoire,
   — à la vérification, à l'aide des différentes variables de compteurs et de la taille de la liste des $\|r_k\|^2$, du nombre d'évaluations des différentes fonctions par itération,
   — à l'affichage des erreurs en fonction du nombre d'itérations, dans une échelle appropriée.
5. Les trois graphiques de la page suivante correspondent aux erreurs $\|r_k\|$ pour cette méthode, et aux normes des gradients de f (où $f(x) = \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle$) aux points correspondant à la méthode du gradient conjugué standard (sans préconditionnement), ou à la méthode de descente de gradient à pas optimal pour la fonction f. Attribuer à chaque graphique la méthode correspondante, en justifiant vos choix. Expliquer l'allure de ces graphiques, en particulier pourquoi l'erreur augmente initialement.
6. * Montrer que si $\delta = 0$, alors la matrice A obtenue à la première question n'est pas définie positive. Montrer que si $\delta > 0$ elle est définie positive.







2 Précision de la méthode de Newton

On suppose que l'on a une fonction f que l'on cherche à minimiser par une méthode de Newton. On va montrer que l'essentiel pour avoir une bonne précision de la méthode est d'avoir une bonne approximation du gradient, et que l'approximation de la hessienne peut être bien moins précise.

On se donne une fonction f de $\mathbb{R}^n$ dans $\mathbb{R}$, de classe C^2, on note H_f sa hessienne. On suppose qu'il existe $x_* \in \mathbb{R}^n$ tel que $\nabla f(x_*) = 0$, et que pour tout x tel que $\|x - x_*\| \leqslant 1$ on ait les deux propriétés :

(i) pour tout $h \in \mathbb{R}^n$, on a $\langle h, H_f(x)h\rangle \geqslant \|h\|^2$;

(ii) pour tout y sur le segment joignant x_* à x, on a $\Big\|[H_f(x) - H_f(y)](x - y)\Big\| \leqslant \|x - y\|^2$.

On se donne enfin x_0 tel que $\|x_0 - x_*\| \leqslant 1$, et on note (x_k) la suite de points obtenus en effectuant la méthode de Newton à pas fixe partant de x_0.

1. Rappeler la formule de récurrence définissant x_{k+1} en fonction de x_k. Montrer que si $\|x - x_*\| \leqslant 1$, alors $H_f(x)$ est inversible et pour tout $h \in \mathbb{R}^n$, on a $\|H_f^{-1}(x)h\| \leqslant \|h\|$. Montrer que le point critique x_* est un point de minimum local strict de f. Est-ce un minimum global ?
2. Montrer que si $\|x - x_*\| \leqslant 1$, alors $\|\nabla f(x_*) - \nabla f(x) - H_f(x)(x_* - x)\| \leqslant \frac{1}{2}\|x - x_*\|^2$.
   Montrer que l'on a alors pour tout $k \leqslant 1$, $\|x_k - x_*\| \leqslant 1$ et $\|x_{k+1} - x_*\| \leqslant \frac{1}{2}\|x_k - x_*\|^2$.

On se donne $\hat{g}(x)$ et $h \mapsto \hat{N}(x, h)$ des approximations de $\nabla f(x)$ et de la fonction correspondant au calcul de la direction de descente de Newton $h \mapsto H_f^{-1}(x)h$, et on suppose que leurs précisions respectives sont $\eta_g \ll 1$ et $\eta_A \ll 1$: si $\|x - x_*\| \leqslant 1$ et $h \in \mathbb{R}^n$, alors

$$\|\nabla f(x) - \hat{g}(x)\| \leqslant \eta_g \quad \text{et} \quad \|H_f^{-1}(x)h - \hat{N}(x, h)\| \leqslant \eta_A\|h\|.$$

On effectue alors la méthode de Newton à l'aide de ces approximations en partant de $\hat{x}_0$ et posant

$$\hat{x}_{k+1} = \hat{x}_k - \hat{N}(\hat{x}_k, \hat{g}(\hat{x}_k)).$$

3. Montrer qu'il existe $K > 0$ tel que l'on ait $\|\nabla f(x)\| \leqslant K\|x - x_*\|$ dès que $\|x - x_*\| \leqslant 1$. Montrer que l'on a alors, si $\|\hat{x}_k - x_*\| \leqslant 1$:

$$\|\hat{x}_{k+1} - x_*\| \leqslant \tfrac{1}{2}\|\hat{x}_k - x_*\|^2 + K\eta_A\|\hat{x}_k - x_*\| + \eta_g + \eta_A\eta_g.$$

4. En déduire que si $\|\hat{x}_0 - x_*\| < 1$, il existe un k_0 tel que $\|\hat{x}_{k_0} - x_*\| \leqslant \max(4K\eta_A, 2\sqrt{\eta_g}))$, et que tant que $\|\hat{x}_k - x_*\| \geqslant \max(4K\eta_A, 4\sqrt{\eta_g}))$, on a alors $\|\hat{x}_k - x_*\| \leqslant \|\hat{x}_0 - x_*\|^{2^k}$.
   Montrer que si $K\eta_A$ est de l'ordre de grandeur de $\sqrt{\eta_g}$ (ou plus petit que $\sqrt{\eta_g}$), il suffit d'une étape de plus pour atteindre la précision de l'ordre de η_g : $\|\hat{x}_{k_0+1} - x_*\| \leqslant C\eta_g$ avec C une constante de l'ordre de grandeur de un, et que l'on ne peut pas espérer une meilleure précision en utilisant seulement l'estimation de la question précédente.
   Dans ce cas on dit que la précision de la méthode de Newton est donc au moins de l'ordre de grandeur de η_g et que l'on atteint cette précision avec une suite $\hat{x}_k$ qui converge quadratiquement.
5. On suppose qu'on ne dispose que de $\hat{g}$ (toujours de précision η_g) et on veut construire $\hat{N}$ par différences finies. On approxime donc $\partial_i\partial_j f$ par différence finies de $\hat{g}_i$. On se donne un paramètre d'approximation ε et on pose $\hat{h}_{ij}(x) = \frac{1}{\varepsilon}[\hat{g}_j(x+\varepsilon e_i) - \hat{g}_j(x)]$, pour $1 \leqslant i \leqslant j \leqslant n$, et $\hat{h}_{ij}(x) = \hat{h}_{ji}(x)$ pour $1 \leqslant j < i \leqslant n$.
   Montrer que si $\partial_i\partial_j f$ est L-Lipschitzienne, on a $|\frac{1}{\varepsilon}[\partial_j f(x + \varepsilon e_i) - \partial_j f(x)] - \partial_i\partial_j f(x)| \leqslant \frac{L}{2}\varepsilon$. En déduire que l'on a, quelque soit $1 \leqslant i, j \leqslant n$:

$$|\hat{h}_{ij}(x) - \partial_i\partial_j f(x)| \leqslant \frac{2\,\eta_g}{\varepsilon} + \frac{L\,\varepsilon}{2}.$$

   Quel est le meilleur choix d'ordre de grandeur de ε pour obtenir la meilleure précision possible pour $\hat{h}_{ij}(x)$ (et quel est alors l'ordre de grandeur de cette précision) ? Quel est le nombre d'évaluations des fonctions $\hat{g}_i$ pour calculer la Hessienne approchée $\hat{H}(x) = (\hat{h}_{ij}(x))$?
   On suppose qu'une fois que l'approximation $\hat{H}(x)$ a été calculée, le calcul de $\hat{H}(x)^{-1}h$ est effectué par une méthode directe d'une précision meilleure que η_g. Pourquoi ne choisit-on pas de prendre l'approximation des différences finies centrées en posant $\hat{h}_{ij}(x) = \frac{1}{2\varepsilon}[\hat{g}_j(x + \varepsilon e_i) - \hat{g}_j(x - \varepsilon e_i)]$, alors que la précision est meilleure (de l'ordre de $(\eta_g)^{2/3}$) ?

3 Taux de convergence de la méthode de gradient à pas fixe pour la norme $\|\cdot\|_A$

On veut minimiser la fonction f de $\mathbb{R}^n$ dans $\mathbb{R}$ donnée par $f(x) = \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle$, où A est une matrice symétrique définie positive et où b est un vecteur de $\mathbb{R}^n$ par la méthode du gradient à pas fixe α.

1. Écrire la relation de récurrence définissant les itérées x_k de la méthode. En notant $r_k = \nabla f(x_k)$, montrer qu'on a $r_{k+1} = r_k - \alpha A r_k$.
2. Montrer que la fonction f admet un unique minimum $x_* \in \mathbb{R}^n$, et que l'on a pour tout k :

$$f(x_k) - f(x_*) = \tfrac{1}{2}\langle x_k - x_*, A(x_k - x_*)\rangle = \tfrac{1}{2}\langle r_k, A^{-1}r_k\rangle.$$

3. On note $\varepsilon_k = f(x_k) - f(x_*)$. On note $(e_1, \ldots, e_n)$ une base orthonormale de vecteurs propres de la matrice A associée aux valeurs propres $(\lambda_1, \ldots, \lambda_n)$. On décompose r_k dans cette base en écrivant $r_k = \sum_{i=1}^n c_{i,k}e_i$.
   Calculer ε_k en fonction des $c_{i,k}$. En déduire que l'on a

$$\varepsilon_{k+1} \leqslant \max_{1\leqslant i\leqslant n} |1 - \alpha\lambda_i|^2\varepsilon_k = \rho(\alpha)^2\varepsilon_k,$$

   où on a posé $\rho(\alpha) = \max(|1 - \alpha\ell|, |1 - \alpha L|)$, avec ℓ (resp. L) la plus petite (resp. la plus grande) valeur propre de A. Montrer qu'il existe des conditions initiales pour lesquelles on a égalité dans l'inégalité précédente quel que soit k.
4. En déduire une majoration du taux de convergence linéaire de x_k vers x_* en norme $\|\cdot\|_A$ (définie par $\|x\|_A^2 = \langle x, Ax\rangle$), lorsque α est dans un certain intervalle que l'on précisera. Quel est le meilleur choix de α rendant ce taux le plus petit possible (le démontrer) ?
5. Comparer ces résultats à ceux pour la convergence de la méthode de gradient à pas fixe pour la norme euclidienne et à ceux pour la convergence de la méthode de gradient à pas optimal.