Optimisation algorithmique
Examen final du 10 janvier 2020 - Correction

Exercice 1. On considère la fonction suivante de $\mathbb{R}^2$ dans $\mathbb{R}$:

$$f(x) = (x_1^2 + x_2^2 - 1)^2.$$

1. Déterminer les points critiques de f.

Correction : Le gradient de f s'écrit

$$\nabla f(x) = \begin{pmatrix} 4(x_1^2 + x_2^2 - 1)x_1 \\ 4(x_1^2 + x_2^2 - 1)x_2 \end{pmatrix} = 4(x_1^2 + x_2^2 - 1)x$$

donc les points critiques de f sont les solutions de

$$(x_1^2 + x_2^2 - 1)x = 0 \quad \Leftrightarrow \quad x_1^2 + x_2^2 - 1 = 0 \text{ ou } x = 0$$

L'équation $x_1^2 + x_2^2 - 1 = 0$ est l'équation du cercle unité. Par conséquent les points critiques de sont : tous les points du cercle unité, ainsi que le point $x = 0$.

---

2. Déterminer le type de chaque point critique (minimiseur local, maximiseur local ou point selle).

Correction : Calculons la matrice hessienne de f. On a

$$D\nabla f(x).h = 4(2x_1h_1+2x_2h_2)x+4(x_1^2+x_2^2-1)h = 8(x^Th)x+4(x_1^2+x_2^2-1)h = (8xx^T+4(x_1^2+x_2^2-1)I)h,$$

donc $Hf(x) = 8xx^T + 4(x_1^2 + x_2^2 - 1)I$.

- Pour tout point du cercle unité, $x_1^2 + x_2^2 - 1 = 0$ donc $Hf(x) = 8xx^T$ qui est une matrice de rang 1 et possède donc une valeur propre nulle. Par conséquent l'analyse de la matrice hessienne ne permet pas de conclure ; il faut procéder différemment. Or si $x_1^2 + x_2^2 - 1 = 0$, on a $f(x) = 0$, et comme f est une fonction positive (c'est un carré), tous ces points critiques sont des minimiseurs globaux de f, et donc aussi des minimiseurs locaux.
- Pour le point critique $x = 0$, la hessienne vaut $Hf(x) = -4I$; ses deux valeurs propres sont strictement négatives et donc ce point est un maximiseur local.

---

3. Quels sont les extrema globaux de f ?

Correction : On a déjà montré que tous les points du cercle unité sont des minimiseurs globaux de f. Il reste à voir si 0 est un maximiseur global. Ceci est clairement faux : on peut par exemple voir que $f(2,0) = (4+0-1)^2 = 9 > 1 = f(0)$.

Exercice 2. Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction deux fois différentiable. On considère une itération d'un algorithme de descente pour la minimisation de la fonction f, consistant, à partir d'une position courante $x \in \mathbb{R}^n$, à choisir une direction de descente $d \in \mathbb{R}^n$, un pas de descente $t > 0$, et à calculer la position à l'itération suivante $x + td$.

1. Rappeler l'expression de la direction d dans le cas de la méthode de Newton.

Correction : $d = -Hf(x)^{-1}\nabla f(x)$.

---

2. Toujours pour la méthode de Newton, montrer que si f est fortement convexe, alors la direction d est bien définie, et qu'il s'agit bien d'une direction de descente, c'est-à-dire que si t est suffisamment petit, alors $f(x+td) < f(x)$.

Correction : Soit $g : \mathbb{R} \to \mathbb{R}$ la fonction définie par $g(t) = f(x+td)$. On a

$$g'(t) = \langle\, \nabla f(x+td)\ ,\ d\,\rangle = -\big\langle\, \nabla f(x+td)\ ,\ Hf(x)^{-1}\nabla f(x)\,\big\rangle,$$

et donc

$$g'(0) = -\big\langle\, \nabla f(x)\ ,\ Hf(x)^{-1}\nabla f(x)\,\big\rangle.$$

f étant fortement convexe, il existe $\alpha > 0$ tel que pour tous $x, h \in \mathbb{R}^n$, $\langle\, Hf(x)h\ ,\ h\,\rangle \geqslant \alpha\|h\|^2$. En particulier $Hf(x)$ est une matrice symétrique définie positive pour tout x ; elle est donc inversible et son inverse $Hf(x)^{-1}$ est également une matrice symétrique définie positive. On a donc $\big\langle\, Hf(x)^{-1}h\ ,\ h\,\big\rangle > 0$ pour tous $x, h \in \mathbb{R}^n$ et donc en prenant $h = \nabla f(x)$,

$$g'(0) = -\big\langle\, \nabla f(x)\ ,\ Hf(x)^{-1}\nabla f(x)\,\big\rangle < 0.$$

La fonction g est donc strictement décroissante en $t = 0$, et donc pour t suffisamment petit on a $g(t) < g(0)$, c'est-à-dire $f(x + td) < f(x)$.

remarque : En fait on n'a utilisé la forte convexité que pour en déduire que $Hf(x)$ est symétrique définie positive. On aurait donc pu remplacer l'hypothèse de forte convexité par : $Hf(x)$ symétrique définie positive pour tout x (ce qui implique que f est strictement convexe mais pas nécessairement fortement convexe).

Exercice 3. Soient k, n deux entiers supérieurs ou égaux à 1, A une matrice réelle de taille (k, n), et $b \in \mathbb{R}^k$. Le problème des moindres carrés linéaire consiste à minimiser la fonction $f : \mathbb{R}^n \to \mathbb{R}$ définie par

$$f(x) = \|Ax - b\|^2,$$

où $\|\cdot\|$ désigne la norme euclidienne canonique sur $\mathbb{R}^k$.

1. Donner les expressions du gradient et de la matrice hessienne de f.

Correction : On a, pour tous $x, h \in \mathbb{R}^n$,

$$Df(x).h = 2\langle\, Ax - b\ ,\ Ah\,\rangle = 2\big\langle\, A^T(Ax - b)\ ,\ h\,\big\rangle,$$

et par ailleurs on sait que $Df(x).h = \langle\, \nabla f(x)\ ,\ h\,\rangle$, donc par identification,

$$\nabla f(x) = 2A^T(Ax - b) = 2A^TAx - 2A^Tb.$$

Ensuite on a

$$D(\nabla f)(x).h = 2A^TAh,$$

et comme on sait que $D(\nabla f)(x).h = Hf(x)h$, on en déduit par identification que

$$Hf(x) = 2A^TA.$$

---

2. On note $F = Im(A) = \{Ax, x \in \mathbb{R}^n\}$. En considérant la décomposition du vecteur b suivant F et $F^\perp$ (sous-espace orthogonal), montrer que f admet au moins un point critique.

Correction : Les points critiques de f sont les solutions de

$$\nabla f(x) = 0 \quad \Leftrightarrow \quad A^T Ax = A^T b.$$

Notons $b = b_F + b_{F^\perp}$ la décomposition de b suivant F et $F^\perp$. On a $b_F \in F = Im(A)$, donc il existe $x \in \mathbb{R}^n$ tel que $b_F = Ax$. Ainsi $b = Ax + b_{F^\perp}$, et donc

$$A^T b = A^T Ax + A^T b_{F^\perp}.$$

Or $b_{F^\perp} \in F^\perp = Im(A)^\perp = Ker(A^T)$, donc $A^T b_{F^\perp} = 0$, et donc $A^T b = A^T Ax$. Le point x ainsi défini est donc un point critique de f.

---

3. Montrer que f est convexe. En déduire que f admet au moins un minimiseur.

Correction : On a $Hf(x) = 2A^T A$. Or, pour tout $h \in \mathbb{R}^n$,

$$\langle\, A^T Ah \,,\, h \,\rangle = \langle\, Ah \,,\, Ah \,\rangle = \|Ah\|^2 \geqslant 0.$$

La matrice hessienne de f est donc symétrique positive en tout point de $\mathbb{R}^n$, ce qui équivaut à la convexité de f.

On sait de plus que pour une fonction convexe différentiable, tous les points critiques sont des minimiseurs globaux. Comme on a montré que f admet un point critique, on en déduit qu'elle admet un minimiseur global.

Application. On cherche à minimiser la fonction $f : \mathbb{R}^2 \to \mathbb{R}$ définie par

$$f(x) = \int_0^1 (t^3 - x_1 t - x_2)^2 dt.$$

4. Montrer qu'il s'agit d'un problème des moindres carrés (on explicitera la matrice A et le vecteur b)

Correction : Montrons d'abord qu'il s'agit d'un problème des moindres carrés : considérons l'espace $\mathbb{R}_3[t]$ des polynômes de degré au plus 3 (espace de dimension 4), muni du produit scalaire

$$\langle\, P \,,\, Q \,\rangle_{\mathbb{R}_3[t]} = \int_0^1 P(t)Q(t)dt.$$

L'expression de $f(x)$ correspond alors à $\|P - Q\|^2_{\mathbb{R}_3[t]}$ où $P(t) = t^3$ et $Q(t) = x_1 t + x_2$. Soit alors $E = Vect(1, t, t^3) \subset \mathbb{R}_3[t]$, sous-espace de dimension 3, et considérons une base orthonormée quelconque (e_0, e_1, e_2) de E. En notant $A \in \mathcal{M}_{3,2}(\mathbb{R})$ la matrice de l'application de $\mathbb{R}^2$ dans

E, qui envoie $x = (x_1, x_2)$ sur $Q(t) = x_1 t + x_2$, écrite pour la base canonique de $\mathbb{R}^2$ et pour la base (e_0, e_1, e_2) de E, et $b = (b_0, b_1, b_2) \in \mathbb{R}^3$ le vecteur des coefficients de la décomposition de t^3 dans la base (e_0, e_1, e_2), on aura exactement que

$$f(x) = \|Ax - b\|^2,$$

ce qui prouve qu'il s'agit d'un problème des moindres carrés.

Pour obtenir explicitement A et b, il faut déterminer la base orthonormée (e_0, e_1, e_2), ce qui est un peu fastidieux. Cependant on peut remarquer qu'il n'était pas nécessaire d'expliciter A et b pour résoudre le problème (i.e. répondre à la question suivante), car on a vu que la solution est la solution du système linéaire $A^T Ax = A^T b$, et les expressions de $A^T A$ et de $A^T b$ peuvent être obtenues directement. En effet on a

$$\begin{aligned}
f(x) &= \int_0^1 (t^3 - x_1 t - x_2)^2 dt = \int_0^1 t^6 + x_1^2 t^2 + x_2^2 - 2x_1 t^4 - 2x_2 t^3 + 2x_1 x_2 t dt \\
&= \left[\frac{t^7}{7} + x_1^2 \frac{t^3}{3} + x_2^2 t - 2x_1 \frac{t^5}{5} - 2x_2 \frac{t^4}{4} + 2x_1 x_2 \frac{t^2}{2} \right]_0^1 \\
&= \frac{1}{7} + x_1^2 \frac{1}{3} + x_2^2 - 2x_1 \frac{1}{5} - 2x_2 \frac{1}{4} + 2x_1 x_2 \frac{1}{2} \\
&= \frac{1}{3} x_1^2 + x_2^2 + x_1 x_2 - \frac{2}{5} x_1 - \frac{1}{2} x_2 + \frac{1}{7} \\
&= \langle\, Mx \,,\, x \,\rangle + \langle\, v \,,\, x \,\rangle + c,
\end{aligned}$$

avec

$$M = \begin{pmatrix} \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix}, \quad v = \begin{pmatrix} -\frac{2}{5} \\ -\frac{1}{2} \end{pmatrix}, \quad c = \frac{1}{7}.$$

et par conséquent en identifiant cette expression avec

$$\|Ax - b\|^2 = \langle\, A^T Ax \,,\, x \,\rangle + \langle\, -2A^T b \,,\, x \,\rangle + \|b\|^2,$$

on voit que $A^T A = M$ et $A^T b = -\frac{1}{2} v$, qui sont donc explicites et permettent de résoudre le problème.

Cependant comme le calcul de A et b était demandé, voici comment faire : on cherche une base orthonormée (e_0, e_1, e_2) de E par un procédé d'orthonormalisation de la base $(1, t, t^3)$: on a d'abord $e_0 = 1$ car $\|1\|^2_{\mathbb{R}_3[t]} = \int_0^1 1 dt = 1$. Ensuite,

$$\langle\, t \,,\, e_0 \,\rangle_{\mathbb{R}_3[t]} = \int_0^1 t dt = \frac{1}{2},$$

$$\left\| t - \frac{1}{2} e_0 \right\|^2_{\mathbb{R}_3[t]} = \|t\|^2_{\mathbb{R}_3[t]} - \frac{1}{4} = \int_0^1 t^2 dt - \frac{1}{4} = \frac{1}{3} - \frac{1}{4} = \frac{1}{12},$$

et donc $e_1 = \sqrt{12}(t - \frac{1}{2}) = 2\sqrt{3}\left(t - \frac{1}{2}\right)$, et t se décompose ainsi :

$$t = \frac{1}{2} e_0 + \frac{\sqrt{3}}{6} e_1.$$

On peut ici déjà écrire la matrice A : il s'agit de la matrice de l'application linéaire de $\mathbb{R}^2$ dans E qui envoie $x = (x_1, x_2)$ sur $x_1 t + x_2$, écrite dans les bases $((1, 0), (0, 1))$ (base canonique de

$\mathbb{R}^2$) et (e_0, e_1, e_2). Cette application envoie $(1,0)$ sur $t = \frac{1}{2}e_0 + \frac{\sqrt{3}}{6}e_1$ et $(0,1)$ sur $1 = e_0$, et par conséquent sa matrice vaut

$$A = \begin{pmatrix} \frac{1}{2} & 1 \\ \frac{\sqrt{3}}{6} & 0 \\ 0 & 0 \end{pmatrix}$$

Ensuite pour trouver b il faut trouver e_2 et décomposer t^3. On a

$$\left\langle t^3 \, , \, e_0 \right\rangle_{\mathbb{R}_3[t]} = \int_0^1 t^3 dt = \frac{1}{4},$$

$$\left\langle t^3 \, , \, e_1 \right\rangle_{\mathbb{R}_3[t]} = 2\sqrt{3}\int_0^1 t^3\left(t - \frac{1}{2}\right)dt = 2\sqrt{3}\int_0^1 \left(t^4 - \frac{1}{2}t^3\right)dt = 2\sqrt{3}\left(\frac{1}{5} - \frac{1}{8}\right) = \frac{3\sqrt{3}}{20},$$

$$\begin{aligned} \left\| t^3 - \frac{1}{4}e_0 - \frac{3\sqrt{3}}{20}e_1 \right\|^2_{\mathbb{R}_3[t]} &= \|t^3\|^2_{\mathbb{R}_3[t]} - \frac{1}{16} - \frac{27}{400} \\ &= \int_0^1 t^6 dt - \frac{1}{16} - \frac{27}{400} \\ &= \frac{1}{7} - \frac{1}{16} - \frac{27}{400} \\ &= \frac{9}{700} \end{aligned}$$

Donc $e_2 = \frac{700}{9}(t^3 - \frac{1}{4}e_0 - \frac{3\sqrt{3}}{20}e_1)$, et t^3 se décompose ainsi :

$$t^3 = \frac{1}{4}e_0 + \frac{3\sqrt{3}}{20}e_1 + \frac{9}{700}e_2.$$

Ainsi finalement,

$$A = \begin{pmatrix} \frac{1}{2} & 1 \\ \frac{\sqrt{3}}{6} & 0 \\ 0 & 0 \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} \frac{1}{4} \\ \frac{3\sqrt{3}}{20} \\ \frac{9}{700} \end{pmatrix}.$$

---

5. Résoudre ce problème.

Correction
On a vu précédemment que les solutions du problème sont les solutions de $A^TAx = A^Tb$. Ici on a

$$A^TA = M = \begin{pmatrix} \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix}$$

Cette matrice est inversible (son déterminant vaut $\frac{1}{3} - \frac{1}{4} \neq 0$), et par conséquent on aura une solution unique. Ensuite,

$$A^Tb = -\frac{1}{2}v = \begin{pmatrix} \frac{1}{5} \\ \frac{1}{4} \end{pmatrix}$$

Le système à résoudre s'écrit donc

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{5} \\ \frac{1}{4} \end{pmatrix} \quad \Leftrightarrow \quad \left\{ \begin{array}{l} \frac{1}{3}x_1 + \frac{1}{2}x_2 = \frac{1}{5} \\ \frac{1}{2}x_1 + x_2 = \frac{1}{4}, \end{array} \right.$$
$$\Leftrightarrow \quad \left\{ \begin{array}{l} \left(\frac{2}{3} - \frac{1}{2}\right) x_1 = \frac{2}{5} - \frac{1}{4} \\ \frac{1}{2}x_1 + x_2 = \frac{1}{4}, \end{array} \right.$$
$$\Leftrightarrow \quad \left\{ \begin{array}{l} \frac{1}{6}x_1 = \frac{3}{20} \\ x_2 = \frac{1}{4} - \frac{1}{2}x_1, \end{array} \right.$$
$$\Leftrightarrow \quad \left\{ \begin{array}{l} x_1 = \frac{9}{10} \\ x_2 = \frac{1}{4} - \frac{1}{2}\frac{9}{10} = \frac{1}{4} - \frac{9}{20} = -\frac{1}{5}, \end{array} \right.$$

La solution du problème est donc

$$x^* = \left(\frac{9}{10}, -\frac{1}{5}\right).$$

Exercice 4. Soit $n \geqslant 1$ un entier, $u : \mathbb{R}^n \to \mathbb{R}$ une fonction deux fois différentiable, A une matrice carrée de taille n, et $y \in \mathbb{R}^n$ un vecteur fixé. On définit $f : \mathbb{R}^n \to \mathbb{R}$ par

$$f(x) = u(Ax) + \|x - y\|^2,$$

où $\|\cdot\|$ désigne la norme euclidienne canonique sur $\mathbb{R}^n$.

1. Donner l'expression de $\nabla f(x)$, gradient de f en un point x, en fonction de A, x, y, et ∇u.

Correction : On a pour tous $x, h \in \mathbb{R}^n$,

$$\begin{aligned} Df(x).h &= Du(Ax).Ah + 2\langle\, x - y \,,\, h \,\rangle = \langle\, \nabla u(Ax) \,,\, Ah \,\rangle + 2\langle\, x - y \,,\, h \,\rangle \\ &= \langle\, A^T \nabla u(Ax) \,,\, h \,\rangle + 2\langle\, x - y \,,\, h \,\rangle \\ &= \langle\, A^T \nabla u(Ax) + 2(x - y) \,,\, h \,\rangle \end{aligned}$$

et donc

$$\nabla f(x) = A^T \nabla u(Ax) + 2(x - y).$$

---

2. Écrire une fonction Python `descente_gradient(u,grad_u,A,y,eta,eps)` qui applique la méthode de descente de gradient sur la fonction f avec pas constant $\eta > 0$ et critère d'arrêt $\|\nabla f(x)\| < \varepsilon$, et renvoie l'approximation du minimiseur de f obtenue. Ici les arguments d'entrée `u, grad_u` sont des fonctions Python supposées déjà codées calculant u et ∇u, et `y` est un vecteur NumPy.

Correction

```
def descente_gradient(u,grad_u,A,y,eta,eps):

    # On définit une fonction pour calculer le gradient
    # (pas indispensable mais ça clarifie le code qui suit) :
    def grad_f(x):
        return A.T @ grad_u(A@x) + 2*(x-y)

    # On initialise par x=0
    x = np.zeros(y.shape)

    # Pour éviter de calculer deux fois par itération le gradient, on l'enregistre :
    gradfx = grad_f(x)

    # boucle principale
    while np.linalg.norm(gradfx) > eps:
        # descente de gradient
        x -= eta * gradfx
        # mise à jour du gradient
        gradfx = grad_f(x)

    return x
```

---

3. Donner l'expression de $Hf(x)$, matrice hessienne de f en un point x, en fonction de A, x, y, et Hu.

Correction : On a pour tous $x, h \in \mathbb{R}^n$,

$$D\nabla f(x).h = A^T D\nabla u(Ax).Ah + 2h = A^T Hu(Ax)Ah + 2h,$$

et donc

$$Hf(x) = A^T Hu(Ax)A + I_n.$$

---

4. Écrire une fonction Python `descente_Newton(u,grad_u,hess_u,A,y,eta,eps)` qui applique la méthode de Newton sur la fonction f avec pas constant $\eta > 0$ et critère d'arrêt $\|\nabla f(x)\| < \varepsilon$. `hess_u` est une fonction Python calculant Hu.

Correction

```
def descente_Newton(u,grad_u,hess_u,A,y,eta,eps):

    n = y.size

    # On définit des fonctions pour calculer le gradient et la hessienne
    def grad_f(x):
        return A.T @ grad_u(A@x) + 2*(x-y)

    def hess_f(x):
        return A.T @ hess_u(A@x) @ A + 2 * np.eye(n)

    # On initialise par x=0
    x = np.zeros(y.shape)

    # Pour éviter de calculer deux fois par itération le gradient, on l'enregistre :
    gradfx = grad_f(x)

    # boucle principale
    while np.linalg.norm(gradfx) > eps:
        # descente de Newton
        x -= eta * np.linalg.solve(hess_f(x),gradfx)
        # mise à jour du gradient
        gradfx = grad_f(x)

    return x
```

---

5. Montrer que si u est convexe, alors f est fortement convexe.

Correction : Si u est convexe, alors sa matrice hessienne $Hu(x)$ est symétrique positive en tout point : pour tous $x, h \in \mathbb{R}^n$ on a $\langle\, Hu(x)h \,,\, h \,\rangle \geqslant 0$. En écrivant cette inégalité pour Ax et Ah au lieu de x et h, on obtient

$$\langle\, Hu(Ax)Ah \,,\, Ah \,\rangle = \langle\, A^T Hu(Ax)Ah \,,\, h \,\rangle \geqslant 0,$$

et donc

$$\langle\, Hf(x)h \,,\, h \,\rangle = \langle\, A^T Hu(Ax)Ah \,,\, h \,\rangle + 2\|h\|^2 \geqslant 2\|h\|^2,$$

ce qui prouve que f est fortement convexe (de constante $m = 2$).

6. Montrer que si de plus il existe une constante $\eta > 0$ telle que

$$\forall x, h \in \mathbb{R}^n, \qquad \langle\, h\ ,\ Hu(x)h\,\rangle \leqslant \eta\|h\|^2,$$

alors il existe deux constantes m, M avec $0 < m < M$ telles que

$$\forall x, h \in \mathbb{R}^n, \qquad m\|h\|^2 \leqslant \langle\, h\ ,\ Hf(x)h\,\rangle \leqslant M\|h\|^2. \quad (*)$$

Correction : L'inégalité de gauche correspond à la forte convexité qui vient d'être démontrée. Ensuite, pour tous $x, h \in \mathbb{R}^n$,

$$\langle\, h\ ,\ Hf(x)h\,\rangle = \langle\, Hu(Ax)Ah\ ,\ Ah\,\rangle + 2\|h\|^2 \leqslant \eta\|Ah\|^2 + 2\|h\|^2 \leqslant \eta|||A|||h\|^2 + 2\|h\|^2,$$

où $|||A|||$ désigne la norme de A subordonnée à la norme euclidienne. On obtient donc l'inégalité demandée avec $M = \eta|||A||| + 2$.

---

7. Que peut-on conclure de l'inégalité $(*)$ vis-à-vis de la convergence des algorithmes de descente de gradient et de l'algorithme de Newton ?

Correction : L'inégalité $(*)$ est suffisante pour assurer la convergence de l'algorithme de descente de gradient. Pour l'algorithme de Newton, il faudrait montrer également que l'application $x \mapsto Hf(x)$ est lipschitzienne.

---

8. Montrer que l'inégalité $(*)$ est vérifiée lorsque $u(x) = \displaystyle\sum_{i=1}^{n} \sqrt{1 + x_i^2}$.

Correction : D'après ce qui précède, il suffit de montrer que u est convexe et que la constante de η de la question 6 existe. Calculons la matrice hessienne de u : d'abord,

$$\frac{\partial u}{\partial x_i}(x) = \frac{x_i}{\sqrt{1 + x_i^2}},$$

puis

$$\frac{\partial^2 u}{\partial x_i \partial x_j}(x) = \begin{cases} 0 \text{ si } i \neq j \\ \frac{1}{\sqrt{1+x_i^2}} - \frac{x_i^2}{(1+x_i^2)^{\frac{3}{2}}} = \frac{1}{(1+x_i^2)^{\frac{3}{2}}} \text{ si } i = j, \end{cases}$$

et donc $Hfu(x)$ est la matrice diagonale de coefficient diagonaux $\frac{1}{(1+x_i^2)^{\frac{3}{2}}}$ pour $1 \leqslant i \leqslant n$. Ces coefficients sont tous positifs, ce qui montre que u est convexe. De plus il sont tous majorés par 1, ce qui montre que l'inégalité $\langle\, h\ ,\ Hu(x)h\,\rangle \leqslant \eta\|h\|^2$ est vérifiée pour $\eta = 1$.

Voici un code permettant de tester les fonctions précédentes avec ce choix de fonction u et une matrice A donnée :

```
import numpy as np
n = 2
A = np.array([[1.,2.],[-1.,1.]])
y = np.array([[1.],[1.]])
# choix du pas : on a m=2 et M=|||A|||+2 :
m, M = 2, np.linalg.norm(A,2)+2
eta = 0.99*(2*m/M**2)
eps = 1e-5
def u(x):
    return np.sum(np.sqrt(1+x**2))
def grad_u(x):
    return x/np.sqrt(1+x**2)
def hess_u(x):
    return np.diag(((1+x**2)**(-3/2)).flatten())

x = descente_gradient(u,grad_u,A,y,eta,eps)
print(x)

x = descente_Newton(u,grad_u,hess_u,A,y,eta,eps)
print(x)
```