Méthodes numériques : optimisation.
Examen du 13 mai 2015 — Éléments de correction

1 Règle de Wolfe, méthode BFGS et positivité.

1. En dimension 1 les vecteurs et les matrices sont des réels, on a donc $I_1 - \frac{s_k y_k^T}{\langle s_k, y_k\rangle} = 1 - \frac{s_k y_k}{s_k y_k} = 0$, et donc pour $k \geqslant 0$, on a $H_{k+1} = \frac{s_k s_k^T}{\langle s_k, y_k\rangle} = \frac{s_k}{y_k} = \frac{x_{k+1}-x_k}{f'(x_{k+1})-f'(x_k)}$. On obtient donc pour $k \geqslant 1$, comme on a supposé que $\alpha_k = 1$

$$x_{k+1} = x_k - H_k \nabla f(x_k) = x_k - \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})} f'(x_k),$$

ce qui correspond exactement à la formule de la sécante.

2. (a) L'autre condition (la règle d'Armijo) consiste à avoir $f(x_k + \alpha_k d_k) \leqslant f(x_k) + c_1 \alpha_k \langle \nabla f(x_k), d_k\rangle$. Les constantes doivent vérifier $0 < c_1 < c_2 < 1$. La règle d'Armijo permet de s'assurer que la fonction diminue suffisamment à l'étape k. Elle n'empêche par contre pas d'avoir un pas trop petit. C'est la deuxième condition, qui demande à ce que la pente dans la direction d_k se soit suffisamment approchée de zéro, qui permet de s'assurer que le pas n'est pas trop petit. Comme la fonction est C^1, on sait que l'on peut satisfaire la règle de Wolfe dès que f est bornée inférieurement et que d_k est une direction de descente.

   (b) On a

$$\begin{aligned}\langle s_k, y_k\rangle &= \langle x_{k+1} - x_k, \nabla f(x_{k+1}) - \nabla f(x_k)\rangle = \alpha_k \langle d_k, \nabla f(x_{k+1}) - \nabla f(x_k)\rangle \\ &\geqslant \alpha_k (c_2 - 1)\langle d_k, \nabla f(x_k)\rangle = -\alpha_k \underbrace{(c_2-1)}_{<0} \underbrace{\langle H_k \nabla f(x_k), \nabla f(x_k)\rangle}_{>0, \text{ car } H_k \text{ s.d.p et } \nabla f(x_k)\neq 0} > 0\end{aligned}$$

3. On a $\langle v, H_{k+1} v\rangle = v^T H_{k+1} v$.
   On calcule $(I_n - \frac{y_k s_k^T}{\langle s_k, y_k\rangle})v = v - \frac{\langle s_k, v\rangle}{\langle s_k, y_k\rangle} y_k = w$ et de même on obtient $v^T(I_n - \frac{s_k y_k^T}{\langle s_k, y_k\rangle} = v^T - \frac{\langle s_k, v\rangle}{\langle s_k, y_k\rangle} y_k^T = w^T$.
   Donc $\langle v, H_{k+1} v\rangle = w^T H_k w + \frac{v^T s_k s_k^T v}{\langle s_k, y_k\rangle} = w^T H_k w + \frac{\langle v, s_k\rangle^2}{\langle s_k, y_k\rangle}$. Comme $\langle s_k, y_k\rangle > 0$ et que H_k est symétrique définie positive, on obtient que $\langle v, H_{k+1} v\rangle \geqslant 0$. Si $\langle v, H_{k+1} v\rangle = 0$ alors $w^T H_k w = 0$ et $\langle v, s_k\rangle = 0$ (somme de deux termes positifs) donc $w = 0$ et en injectant ceci dans la définition de w on obtient $v = 0$. Donc H_{k+1} est symétrique définie positive.

4. Les deux questions précédentes montrent immédiatement par récurrence (puisque H_0 est symétrique définie positive) que tant que $\nabla f(x_k) \neq 0$, la formule BFGS est bien définie (car $\langle y_k, s_k\rangle > 0$) et donne une matrice H_{k+1} symétrique définie positive.
   Comme on a $(I_n - \frac{y_k s_k^T}{\langle s_k, y_k\rangle})y_k = y_k - y_k \frac{\langle y_k, s_k\rangle}{\langle s_k, y_k\rangle} = 0$, on obtient que $H_{k+1} y_k = \frac{\langle s_k, y_k\rangle}{\langle s_k, y_k\rangle} s_k = s_k$. Et donc $y_k = H_{k+1}^{-1} s_k$ ce qui est la formule demandée.
   On a bien $\nabla f(x_{k+1}) - \nabla f(x_k) = \mathrm{Hess}_f(x_{k+1})(x_{k+1} - x_k) + o(\|x_{k+1} - x_k\|)$ (la hessienne est la différentielle du gradient). C'est en cela que l'on dit que H_{k+1}^{-1} est une approximation de la Hessienne en x_{k+1} puisqu'elle vérifie la même formule sans le $o(\|x_{k+1} - x_k\|)$. On appelle cette méthode quasi-Newton car on choisit $d_k = -H_k \nabla f(x_k)$ où H_k est donc une approximation de l'inverse de la Hessienne, alors que dans la méthode de Newton, on prend exactement l'inverse de la hessienne à la place de H_k.

2 Cas test de la méthode de descente de gradient à pas optimal en dimension deux

1. On sait que $\nabla f(x_*) = Ax_* + B = 0$. On remplace donc B par $-Ax_*$ dans l'expression de $f(x_*)$ et $f(x_k)$, et on obtient

$$\begin{aligned}f(x_k) - f(x_*) &= \frac{1}{2}\langle x_k, Ax_k\rangle - \langle Ax_*, x_k\rangle - \frac{1}{2}\langle x_*, Ax_*\rangle + \langle Ax_*, x_k\rangle \\ &= \frac{1}{2}\langle x_k, Ax_k\rangle - \langle Ax_*, x_k\rangle - \frac{1}{2}\langle x_*, Ax_*\rangle = \frac{1}{2}\langle x_k - x_*, A(x_k - x_*)\rangle,\end{aligned}$$

   par symétrie de A (on a $\langle x_k, Ax_*\rangle = \langle Ax_k, x_*\rangle$). Et comme $r_k = \nabla f(x_k) = Ax_k + B = A(x_k - x_*)$, on remplace $x_k - x_*$ par A^{-1} pour obtenir le résultat demandé.

2. On a $h'(t) = \langle \nabla f(x+td), d\rangle = \langle A(x+td)+B, d\rangle = \langle Ax+B, d\rangle + t\langle Ad, d\rangle$. Comme A est symétrique définie positive, on a $\langle Ad, d\rangle > 0$ puisque $d \neq 0$, et donc la seule solution à $h'(t) = 0$ est $t = \frac{-\langle Ax+B, d\rangle}{\langle d, Ad\rangle}$. La fonction $h'(t)$ est affine avec un coefficient directeur positif, donc croissante, donc h est convexe, le seul point critique est un minimum.

3. On a donc $\alpha_k = \frac{-\langle Ax_k+B, -r_k\rangle}{-r_k, A(-r_k)}$ d'après la question précédente, ce qui donne le résultat voulu puisque $r_k = Ax_k + B$.
On a donc $r_{k+1} = Ax_{k+1} + B = A(x_k - \alpha_k r_k) + B = Ax_k + B - \alpha_k Ar_k = r_k - \alpha_k Ar_k$ ce qui est la formule demandée.
On peut donc calculer $\langle r_{k+1}, r_k\rangle = \langle r_k, r_k\rangle - \alpha_k\langle Ar_k, r_k\rangle$ qui vaut zéro d'après l'expression de α_k.
Ce résultat est valable en fait dès que l'on fait une descente de gradient à pas optimal : on sait que la direction de descente est orthogonale au gradient au point d'arrivée. Donc si la direction de descente est donnée par l'opposé du gradient, on obtient que les gradients consécutifs sont orthogonaux.

4. (a) Si $\ell = L$ alors $A = LI_2$ et $\alpha_0 = \frac{1}{L}$, puis $r_1 = r_0 - \alpha_0 Lr_0 = 0$. Si $a = 0$ alors $Ar_0 = \ell r_0$ donc $\alpha_0 = \frac{1}{\ell}$ et $r_1 = r_0 - \alpha_0 \ell r_0 = 0$, de même si $b = 0$ alors $Ar_0 = Lr_0$ et $\alpha_0 = \frac{1}{L}$ puis $r_1 = r_0 - \alpha_0 Lr_0 = 0$. L'algorithme s'arrête en une étape dans les trois cas.

   (b) Les vecteurs $r_0 = a\,e_L + b\,e_\ell$ et $-b\,e_L + a\,e_\ell$ sont orthogonaux (puisque e_ℓ, e_L est une base orthonormale, on a $\langle a\,e_L + b\,e_\ell, -b\,e_L + a\,e_\ell\rangle = -ab + ba = 0$). Ils forment donc une base. Par récurrence, comme les r_k sont orthogonaux deux à deux, et que l'espace est de dimension 2, on obtient que tant que $r_k \neq 0$, r_{k+1} est colinéaire à l'un ou l'autre des vecteurs de cette base, alternativement pour k pair ou k impair. Si l'un des r_k est nul, alors tous les autres sont nuls aux étapes suivantes et on prend les c_k tous nuls à partir de ce moment-là.

   (c) On a alors $\langle r_k, r_k\rangle = c_k^2(a^2+b^2)$ dans les deux cas puisque la base est e_L, e_ℓ est orthonormale. Et pour k pair (resp. impair), $Ar_k = c_k(La\,e_L + \ell b\,e_\ell)$ (resp. $c_k(-Lb\,e_L + \ell a\,e_\ell)$), donc on obtient $\langle r_k, Ar_k\rangle = c_k^2(La^2 + \ell b^2)$ (resp. $c_k^2(Lb^2 + \ell a^2)$), ce qui donne bien le résultat voulu par la formule $\alpha_k = \frac{\langle r_k, r_k\rangle}{\langle r_k, Ar_k\rangle}$.

   (d) Si k est pair, on a d'après les calculs précédents

$$\begin{aligned} r_{k+1} = r_k - \alpha_k Ar_k &= c_k\left(a\,e_L + b\,e_\ell - \frac{a^2+b^2}{La^2+\ell b^2}(La\,e_L + \ell b\,e_\ell)\right) \\ &= c_k\left(\frac{a(La^2+\ell b^2) - aL(a^2+b^2)}{La^2+\ell b^2}e_L + \frac{b(La^2+\ell b^2) - \ell b(a^2+b^2)}{La^2+\ell b^2}e_\ell\right) \\ &= \frac{c_k}{La^2+\ell b^2}(-ab^2(L-\ell)e_L + a^2b(L-\ell)e_\ell) \\ &= c_k\frac{(L-\ell)ab}{La^2+\ell b^2}(-be_L + ae_\ell). \end{aligned}$$

   Comme $k+1$ est impair, on a $r_{k+1} = c_{k+1}(-be_L + ae_\ell)$, on en déduit que $c_{k+1} = c_k\frac{(L-\ell)ab}{La^2+\ell b^2}$.
   Lorsque k est impair, on fait le même calcul en utilisant les résultats précédents

$$\begin{aligned} r_{k+1} = r_k - \alpha_k Ar_k &= c_k\left(-b\,e_L + a\,e_\ell - \frac{a^2+b^2}{Lb^2+\ell a^2}(-Lb\,e_L + \ell a\,e_\ell)\right) \\ &= c_k\left(\frac{-b(Lb^2+\ell a^2) + bL(a^2+b^2)}{Lb^2+\ell a^2}e_L + \frac{a(Lb^2+\ell a^2) - \ell a(a^2+b^2)}{La^2+\ell b^2}e_\ell\right) \\ &= \frac{c_k}{Lb^2+\ell a^2}(a^2b^2(L-\ell)e_L + ab^2(L-\ell)e_\ell) \\ &= c_k\frac{(L-\ell)ab}{Lb^2+\ell a^2}(ae_L + be_\ell). \end{aligned}$$

   Comme $k+1$ est pair, on a $r_{k+1} = c_{k+1}(ae_L + be_\ell)$, on en déduit que $c_{k+1} = c_k\frac{(L-\ell)ab}{Lb^2+\ell a^2}$.

5. Dans le cas où k est pair, on a $A^{-1}r_k = c_k(\frac{a}{L}e_L + \frac{b}{\ell}e_\ell)$ et donc $\langle r_k, A^{-1}r_k\rangle = c_k^2(\frac{a^2}{L} + \frac{b^2}{\ell}) = c_k^2\frac{\ell a^2 + Lb^2}{\ell L}$. De même si k est impair, on obtient $\langle r_k, A^{-1}r_k\rangle = c_k^2\frac{\ell b^2 + La^2}{\ell L}$.

Si k est pair, on obtient donc

$$\begin{aligned}\langle r_{k+1}, A^{-1}r_{k+1}\rangle &= c_{k+1}^2\frac{\ell b^2+La^2}{\ell L}\\ &= c_k^2\frac{(L-\ell)^2a^2b^2(\ell b^2+La^2)}{(La^2+\ell b^2)^2\ell L}\\ &= c_k^2\frac{\ell a^2+Lb^2}{L\ell}\frac{(L-\ell)^2a^2b^2}{(\ell a^2+Lb^2)(La^2+\ell b^2)}\\ &= \langle r_k, A^{-1}r_k\rangle\frac{(L-\ell)^2a^2b^2}{(La^2+\ell b^2)(\ell a^2+Lb^2)}.\end{aligned}$$

Si k est impair, on obtient également de la même manière.

$$\begin{aligned}\langle r_{k+1}, A^{-1}r_{k+1}\rangle = c_{k+1}^2\frac{\ell a^2+Lb^2}{\ell L} &= c_k^2\frac{(L-\ell)^2a^2b^2(\ell a^2+Lb^2)}{(Lb^2+\ell a^2)^2\ell L}\\ = c_k^2\frac{\ell b^2+La^2}{L\ell}\frac{(L-\ell)^2a^2b^2}{(\ell b^2+La^2)(Lb^2+\ell a^2)} &= \langle r_k, A^{-1}r_k\rangle\frac{(L-\ell)^2a^2b^2}{(La^2+\ell b^2)(\ell a^2+Lb^2)}.\end{aligned}$$

(a) Le taux est $\max(|1-\alpha L|, |1-\alpha\ell|)$, et le meilleur taux est obtenu lorsque $|1-\alpha L| = |1-\alpha\ell|$ qui est équivalent à $\alpha = \frac{2}{L+\ell}$, le taux est alors $\rho_{\text{fixe}} = \frac{L-\ell}{L+\ell}$.

(b) D'après la question 1. on a donc

$$\begin{aligned}\|x_{k+1}-x_*\|_A &= \sqrt{\langle r_{k+1}, A^{-1}r_{k+1}\rangle}\\ &= \frac{(L-\ell)|ab|}{\sqrt{(La^2+\ell b^2)(\ell a^2+Lb^2)}}\sqrt{\langle r_{k+1}, A^{-1}r_{k+1}\rangle}\\ &= \frac{(L-\ell)|ab|}{\sqrt{L^2a^2b^2+\ell^2a^2b^2+L\ell(a^4+b^4)}}\|x_k-x_*\|_A.\end{aligned}$$

En divisant par $|ab|$ au numérateur et au dénominateur (donc en divisant par a^2b^2 sous la racine), on obtient bien $\|x_{k+1}-x_*\|_A = \rho_{\text{opt}}\|x_k-x_*\|_A$, et par une récurrence immédiate on obtient le résultat voulu.

(c) On prend $t = \frac{a^2}{b^2}$, on obtient que $\frac{a^2}{b^2}+\frac{b^2}{a^2} \geqslant 2$ avec égalité si et seulement si $|a| = |b|$.
Et donc $\rho_{\text{opt}} \leqslant \frac{L-\ell}{\sqrt{L^2+\ell^2+2L\ell}} = \frac{L-\ell}{L+\ell}$ avec égalité si et seulement si $|a| = |b|$.

Dans ce cas, la formule de α_k donne $\alpha_k = \frac{2}{L+\ell}$ dans les deux cas, c'est donc une descente à pas fixe ! Cela correspond bien au meilleur pas fixe, et donc au meilleur taux de la méthode de gradient à pas fixe.

3 Gradient conjugué pour les moindres carrés linéaires

1. Le problème revient à savoir s'il existe $x_* \in \mathbb{R}^n$ tel que $\|Mx_*-y\| \leqslant \|Mx-y\|$ pour tout $x \in \mathbb{R}^n$, c'est à dire trouver un $z_* \in \text{Im}(M)$ tel que $\|z_*-y\| \leqslant \|z-y\|$ pour tout $z \in \text{Im}(M)$.
Un tel z_* existe et est unique, c'est la projection orthogonale de y sur le sous-espace vectoriel $\text{Im}(M)$, qui est caractérisé par $z_* \in \text{Im}(M)$ et $z_*-y \in \text{Im}(M)^\perp = \ker M^T$, c'est à dire $M^T(z_*-y) = 0$. Donc les solutions x_* au problème (pas forcément uniques) sont caractérisées par $Mx_* = z_*$, c'est à dire $M^T(Mx_*-y) = 0$.
Une autre manière de voir la caractérisation est en développant $g(x) = \|Mx-y\|^2$ (mais c'est ce qu'on demande plus ou moins de faire à la question suivante), et en calculant le gradient $\nabla g(x) = 2M^TMx - M^Ty$. Comme g est convexe, les solutions de $\nabla g = 0$ sont exactement les minimiseurs de g.

2. Par le théorème du rang, si $\ker M = \{0\}$ alors $\dim\text{Im}(M) = n$, et comme $\text{Im}(M) \subset \mathbb{R}^m$ on a forcément $m \geqslant n$. Comme M est injective, il n'y a qu'une solution à $Mx_* = z_*$, donc la solution au problème est unique.
Autre manière de le voir : M^TM est symétrique définie positive. En effet $\langle x, M^TMx\rangle = \|Mx\|^2 \geqslant 0$, et si $\langle x, M^TMx\rangle = 0$, alors $Mx = 0$ et donc $x = 0$. Donc M^TM est inversible et la solution à l'équation normale est unique.

3. On calcule

$$\|Mx-y\|^2 = \langle Mx, Mx\rangle - 2\langle Mx, y\rangle + \langle y, y\rangle = 2(\langle x, M^TMx\rangle - \langle x, M^Ty\rangle) + \|y\|^2 = 2f(x) + \|y\|^2.$$

Minimiser $\|Mx-y\|$ pour $x \in \mathbb{R}^n$ revient donc à minimiser $f(x)$ pour $x \in \mathbb{R}^n$.
On peut appliquer la méthode de gradient conjugué car la fonction f est quadratique avec $A = M^TM$ symétrique définie positive (voir question précédente). On a intérêt à appliquer la méthode du gradient conjugué lorsque la

dimension est grande et qu'il est trop coûteux de résoudre un système linéaire de taille $n \times n$ par des méthodes directes (comme le pivot de Gauss, ou plutôt ici une décomposition de Cholesky, vu que la matrice est symétrique). En effet, la méthode du gradient conjugué fournira une solution approchée, en ne nécessitant que le calcul du produit entre M^TM et un vecteur p_k à chaque étape (ce qui est bien moins coûteux que de résoudre un système linéaire).

4. On prend x_0, on calcule $r_0 = M^TMx_0 - M^Ty$ et on prend $p_0 = -r_0$. Pour tout $k \geqslant 0$, si $r_k \neq 0$, on pose

$$\begin{cases} \alpha_k = \frac{\|r_k\|^2}{\langle p_k, M^TMp_k\rangle} \\ x_{k+1} = x_k + \alpha_k p_k \\ r_{k+1} = r_k + \alpha_k M^TMp_k \\ p_{k+1} = -r_{k+1} + \frac{\|r_{k+1}\|^2}{\|r_k\|^2} p_k \end{cases}.$$

5. Si $Mx = 0$ alors pour $1 \leqslant i \leqslant n$, $(Mx)_i = x_i d_i = 0$ donc $x_i = 0$ car $d_i \neq 0$. Donc $x = 0$, et $\ker M = \{0\}$. On a $(M^TM)_{ij} = c_ic_j \neq 0$ si $i \neq j$ et $(M^TM)_{ii} = d_i^2 + c_i^2 > 0$. Tous les coefficients sont donc non-nuls.
   Pour calculer $((M^TM)p_k)_i = \sum_{j=1}^n (M^TM)_{ij}(p_k)_j$, cela nécessite donc n multiplications. Comme on doit calculer cela pour $1 \leqslant i \leqslant n$, cela fait en tout n^2 multiplications.
   Pour calculer $(Mp_k)_i$, on a une seule multiplication $d_i(p_k)_i$ si $1 \leqslant i \leqslant n$, et n multiplication pour calculer $(Mp_k)_{n+1} = \sum_{j=1}^n c_j(p_k)_j$. Donc en tout il faut $2n$ multiplications pour calculer $(Mp_k)_i$. Ensuite sur chaque ligne i de M^T, il n'y a que deux éléments non-nuls : d_i et c_i. Donc il n'y a que 2 multiplications à faire pour calculer $(M^T(Mp_k))_i = d_i(Mp_k)_i + c_i(Mp_k)_{n+1}$. Comme on fait le calcul pour $1 \leqslant i \leqslant n$, cela fait donc $2n$ calculs. Au final il a donc fallu seulement $4n$ multiplications pour calculer $M^T(Mp_k)$.
   On a donc tout intérêt à ne pas stocker M^TM et recalculer à chaque fois $M^T(Mp_k)$ dans cet ordre. L'autre raison est une raison de place mémoire : stocker M^TM demande à stocker n^2 réels (en fait $\frac{n(n+1)}{2}$ puisque la matrice est symétrique), ce qui peut être trop gros si n est assez grand, alors que M ne demande que de stocker $2n$ réels.

6. On a $Mp = \begin{pmatrix} \varepsilon \times 1 + (-1+\delta) \times 0 \\ 1 \times 1 + (-1+\delta) \times 1 \end{pmatrix}$.
   La machine fera donc une erreur de l'ordre de $O(\varepsilon\eta)$ pour la première ligne et de l'ordre de $O(\eta)$ pour l'addition de la deuxième ligne, donc elle renverra $\begin{pmatrix} \varepsilon + O(\varepsilon\eta) \\ \delta + O(\eta) \end{pmatrix}$.
   Ensuite pour calculer la norme de ce vecteur au carré, elle effectuera d'abord les deux multiplications $(\varepsilon + O(\varepsilon\eta))^2$ et $(\delta + O(\eta))^2$, et fera donc des erreurs d'ordre $O(\varepsilon^2\eta)$ et $O(\delta\eta)$.
   Elle renverra donc $\varepsilon^2 + 2\varepsilon O(\varepsilon\eta) + O(\varepsilon^2\eta^2) + O(\varepsilon^2\eta) = \varepsilon^2 + O(\varepsilon^2\eta)$ pour le premier produit, et de même elle renvoie $\delta^2 + 2\delta O(\eta) + O(\eta^2) + O(\delta\eta) = \delta^2 + O(\delta\eta)$ pour le deuxième produit. En faisant la somme, il restera des erreurs d'ordre $O(\max(\varepsilon^2, \delta^2)\eta)$, donc elle renvoie $\varepsilon^2 + O(\varepsilon^2\eta) + \delta^2 + O(\max(\varepsilon^2, \delta^2)\eta) + O(\delta\eta)$, c'est-à-dire $\varepsilon^2 + \delta^2 + O((\varepsilon^2 + \delta)\eta)$. L'erreur finale est donc d'ordre $O((\varepsilon^2 + \delta)\eta)$, pour le calcul évaluant d'abord Mp, puis $\|Mp\|^2$.
   Pour le calcul de $M(Mp)$, elle effectue donc $\begin{pmatrix} \varepsilon \times (\varepsilon + O(\varepsilon\eta)) + 1 \times (\delta + O(\eta)) \\ 0 \times (\varepsilon + O(\varepsilon\eta)) + 1 \times (\delta + O(\eta)) \end{pmatrix}$, puisqu'on sait que le calcul de Mp a renvoyé $\begin{pmatrix} \varepsilon + O(\varepsilon\eta) \\ \delta + O(\eta) \end{pmatrix}$, .
   Pour la première ligne, elle fait donc une erreur d'ordre $O(\varepsilon^2\eta)$ pour le calcul du premier produit (obtenant donc $\varepsilon^2 + \varepsilon O(\varepsilon\eta) + O(\varepsilon^2\eta) = \varepsilon^2 + O(\varepsilon^2\eta)$) et une erreur d'ordre $O(\delta\eta)$ pour le deuxième produit (obtenant donc $\delta + O(\eta) + O(\delta\eta) = \delta + O(\eta)$). En faisant la somme elle fait donc une erreur de $O(\max(\varepsilon^2, \delta)\eta)$, et renvoie $\varepsilon^2 + O(\varepsilon^2\eta) + \delta + O(\eta) + O(\max(\varepsilon^2, \delta)\eta) = \varepsilon^2 + \delta + O(\eta)$.
   Pour la deuxième ligne on a déjà fait le raisonnement, elle renvoie $\delta + O(\eta)$. Le résultat renvoyé par le calcul $M(Mp)$ est donc $\begin{pmatrix} \varepsilon^2 + \delta + O(\eta) \\ \delta + O(\eta) \end{pmatrix}$.
   Enfin pour le produit scalaire de ce vecteur avec p, elle fait donc les produits $1 \times (\varepsilon^2 + \delta + O(\eta))$ (avec une erreur $O((\varepsilon^2 + \delta)\eta)$, renvoyant $\varepsilon^2 + \delta + O(\eta) + O((\varepsilon^2 + \delta)\eta) = \varepsilon^2 + \delta + O(\eta)$), et $(-1+\delta) \times (\delta + O(\eta))$ (avec une erreur de $O((1-\delta)\delta\eta) = O(\delta\eta)$, elle renvoie $-\delta + \delta^2 + (-1+\delta)O(\eta) + O((1-\delta)\delta\eta) = -\delta + \delta^2 + O(\eta)$).
   Pour faire la somme de ces deux quantités, elle fait alors une erreur de $O(\max(\varepsilon^2 + \delta, \delta - \delta^2)\eta)$, donc le résultat final renvoyé est $\varepsilon^2 + \delta + O(\eta) - \delta + \delta^2 + O(\eta) + O(\max(\varepsilon^2 + \delta, \delta - \delta^2)\eta) = \varepsilon^2 + \delta^2 + O(\eta)$.
   L'erreur finale est donc d'ordre $O(\eta)$ pour le calcul évaluant Mp, puis $M(Mp)$ puis $\langle p, M(Mp)\rangle$.
   On voit donc que le résultat peut être complètement faux si $\eta > \delta^2 + \varepsilon^2$, alors que dans le cas précédent, si $\delta \ll \eta$, on a toujours une erreur $O((\varepsilon^2 + \delta)\eta)$ qui est petite devant le résultat exact.