Enoncés et corrections : Ana Matos.

Factorisation QR. Transformations de Givens. Moindres carrés

Exercice 1 Matrices de Householder

1. Soit v un vecteur réel vérifiant $v^T v = 1$. Montrer que la matrice de Householder
$$H(v) = I - 2vv^T$$
représente une symétrie par rapport au sous–espace vectoriel formé par les vecteurs orthogonaux aux vecgteurs v. En déduire que $\det(H(v)) = -1$.
2. Démontrer que toute matrice orthogonale est le produit de au plus n matrices de Householder. En déduire une interprétation géométrique des matrices orthogonales.

Correction ▼ [002229]

Exercice 2 Algorithme de Gram–Schmidt et Gram–Schmidt modifié

Etant donnés n vecteurs linéairement indépendants de $\mathbb{R}^m$, $\{a_1, \cdots, a_n\}$, on veut calculer une base orthonormale pour $\mathrm{span}\{a_1, \cdots, a_n\}$.
On pose $A = [a_1, a_2, \cdots, a_n] \in \mathbb{R}^{m\times n}$ et on considère la factorisation QR de A,
$$A = QR, \quad Q = [q_1, \cdots, q_n], \quad r_i^T, i = 1, \cdots, n \text{ les lignes de } R$$

1. Montrer que
$$\mathrm{Im}A = \mathrm{span}\{q_1, \cdots, q_n\}.$$
2. Montrer que
$$q_k = \frac{1}{r_{kk}}\left(a_k - \sum_{i=1}^{k-1} r_{ik} q_i\right) \quad k = 1, \cdots, n$$
3. En déduire un algorithme pour le calcul récursif des q_i (algorithme de Gram–Schmidt).
4. Algorithme de Gram–Schmidt modifié
L'algorithme précédent est instable numériquement dû à la perte d'orthogonalité dans le calcul des q_i. On va reformuler l'algorithme pour le rendre stable.
Pour $k = 1, \cdots, n-1$, on définit $A^{(k)} \in \mathbb{R}^{m\times(n-k+1)}$ de la façon suivante :
$$[0, A^{(k)}] = A - \sum_{i=1}^{k-1} q_i r_i^T = \sum_{i=k}^{n} q_i r_i^T$$
et on va décrire l'étape k de l'algorithme.
   (a) Montrer que si on pose
   $$A^{(k)} = [z, B], \quad z \in \mathbb{R}^m, \quad B \in \mathbb{R}^{m\times(n-k)}$$
   alors
   $$r_{kk} = \|z\|_2, \quad q_k = z/r_{kk}.$$
   (b) Comment peut–on calculer la ligne k de R à partir de $A^{(k)}$?
   (c) Calculer $A^{(k+1)}$.
   (d) A partir des questions précédentes, décrire l'algorithme qui permet le calcul de la factorisation $A = Q_1R_1$, $Q_1 \in \mathbb{R}^{m\times n}$ orthonormale, $R_1 \in \mathbb{R}^{n\times n}$ triangulaire supérieure (Gram–Schmidt modifié). Le calcul de Q_1 doit se faire sur place.
   (e) Quelle est la complexité de l'algorithme précédent ?

Correction ▼ [002230]

Exercice 3 Rotation de Givens

Soient $p,q : 1 \leq p < q \leq n$, $c,s \in \mathbb{R} : c^2+s^2 = 1$.
On considère les matrices

$$G = G_{p,q}(c,s) = \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & c & \cdots & -s & & \\ & & & & \ddots & & & \\ & & & s & \cdots & c & & \\ & & & & & & \ddots & \\ & & \cdots & & & & & 1 \end{pmatrix}$$

1. Ecrire G comme perturbée de I par des matrices de rang 1.
2. Montrer que G est inversible, calculer G^{-1}, montrer que G est orthogonale.
3. Quelle est l'action de G sur $A \in \mathbb{R}^{n\times n}$?
4. Soit $A \in \mathbb{R}^{n\times n}$ avec $a_{pj} = \alpha, a_{qj} = \beta$. Peut–on trouver G telle que $A' = GA$ vérifie :

   $$a'_{pj} = 0 = \alpha', \quad a'_{qj} = 0 = \beta'?$$

   Est–ce que la solution est unique ?

Correction ▼ [002231]

Exercice 4

Soit $Z = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$ avec $c^2+s^2 = 1$. On définit ρ par

$$\rho = \begin{cases} 1 & \text{si} \quad c = 0 \\ 1/2\text{sign}(c)s & \text{si} \quad |s| < |c| \\ 2\text{sign}(s)/c & \text{si} \quad |c| \leq |s| \end{cases}$$

1. Comment reconstruire $\pm Z$ à partir de ρ ?
2. Soit Q une matrice orthogonale produit de n rotations de Givens : $Q = J_1 \cdots J_n$. Comment peut–on stocker de la façon la plus économique Q sous forme factorisée ?
3. Modifier l'algorithme de Givens pour réduire A à la forme triangulaire supérieure ($QA = R$, Q matrice produit de rotations de Givens) en stockant sur place (donc dans A) toute l'information nécessaire à reconstruire Q.
4. Ecrire l'algorithme qui, à partir des résultats de l'algorithme précédent permet de reconstruire Q.

[002232]

Exercice 5

Soient x et y deux vecteurs unitaires. Donner un algorithme qui utilise les transformations de Givens pour calculer une matrice Q telle que $Qx = y$. [002233]

Exercice 6 Méthode de Givens Rapide

Soit $A \in \mathbb{R}^{m\times n}$. On veut construire une matrice $M \in \mathbb{R}^{m\times m}$ telle que
– $MA = S$ triangulaire supérieure ;
– $MM^T = D = \text{diag}(d_1, \cdots, d_m)$, $d_i > 0$

et appliquer cette factorisation de A dans la résolution de systèmes au sens des moindres carrés.

1. Donner la factorisation QR de A en termes de M, D et S.
2. On considère maintenant $m = 2$. Soient $x = (x_1, x_2)^T$ et $D = \text{diag}(d_1, d_2)$ ($d_i > 0$) donnés.
   (a) On définit
   $$M_1 = \begin{pmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{pmatrix}.$$
   Supposons $x_2 \neq 0$. Calculer $M_1 x$ et $M_1 D M_1^T$.
   Comment choisir α_1 et β_1 de façon à ce que la deuxième composante de $M_1 x$ soit nulle et que $M_1 D M_1^T$ soit diagonale ?
   Pour le choix précédent déterminer γ_1 tel que
   $$M_1 x = \begin{pmatrix} x_2(1+\gamma_1) \\ 0 \end{pmatrix} \text{ et } M_1 D M_1^T = \begin{pmatrix} d_2(1+\gamma_1) & 0 \\ 0 & d_1(1+\gamma_1) \end{pmatrix}$$
   (b) Supposons $x_1 \neq 0$. On définit
   $$M_2 = \begin{pmatrix} 1 & \alpha_2 \\ \beta_2 & 1 \end{pmatrix}.$$
   Choisir α_2 et β_2 de façon à ce que
   $$M_2 x = \begin{pmatrix} x_1(1+\gamma_2) \\ 0 \end{pmatrix} \text{ et } M_2 D M_2^T = \begin{pmatrix} d_1(1+\gamma_2) & 0 \\ 0 & d_2(1+\gamma_2) \end{pmatrix}$$
   et déterminer γ_2
   (c) Montrer que l'on peut toujours choisir M_i ($i = 1,2$) de façon à ce que le "facteur de croissance" $(1+\gamma_i)$ soit inférieur à 2.
3. Soit maintenant $m \in \mathbb{N}$ quelconque. Définir les matrices $M_1(p,q)$ et $M_2(p,q)$ telles que
   $$\begin{pmatrix} m_{pp} & m_{pq} \\ m_{qp} & m_{qq} \end{pmatrix} = \begin{pmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{pmatrix} \text{ ou } = \begin{pmatrix} 1 & \alpha_2 \\ \beta_2 & 1 \end{pmatrix}$$
   – $e_q^T M_i(p,q) x = 0$;
   – $M_i D M_i^T$ matrice diagonale, avec $D = \text{diag}(d_1, \cdots, d_n), d_i > 0$
   Ces matrices M_i sont appelées *matrice de Givens rapide*.
4. Décrire l'algorithme qui utilise les transformations de Givens rapides pour réduire $A \in \mathbb{R}^{m \times n}$ à la forme triangulaire supérieure (*méthode de Givens rapide*) :
   $$MA = R, \quad MM^T = \text{diag}(d_1, \cdots, d_m).$$
   Les calculs doivent être faits sur place.
   Quel est le coût de cet algorithme ? Comparer avec le coût de la méthode de Householder pour réduire A à la forme triangulaire supérieure.
5. Application à la résolution d'un système linéaire au sens des moindres carrés.
   (a) Comment profiter des résultats fournis par l'algorithme précédent pour résoudre
   $$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2 \text{ avec } A \in \mathbb{R}^{m \times n} \quad (m > n), \quad b \in \mathbb{R}^m ?$$
   (b) Quelles modifications introduire dans l'algorithme de la méthode de Givens rapide pour qu'il résolve le problème de moindres carrés de la question précédente ?
6. *Application numérique* : résoudre au sens des moindres carrés par la méthode de Givens rapide le système
   $$Ax = b, A = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}, b = \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}$$

7. Considérons maintenant le problème de moindres carrés

$$\min_{x\in\mathbb{R}^n} \|D(Ax-b)\|_2 \tag{1}$$

avec $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $D =$diag(d_i) $(d_i > 0)$. Cela correspond à donner un poids différent à chaque équation du système.
Soit M une matrice produit de matrices de Givens rapide vérifiant

$$\begin{cases} MA = R \text{ triangulaire supérieure} \\ MD^{-2}M^T = \tilde{D} = \mathrm{diag}(\tilde{d_i}), \quad \tilde{d_i} > 0 \end{cases}$$

Comment peut–on résoudre le problème (1) ?
Quelles adaptations faire à l'algorithme précédent ?

Correction ▼ [002234]

Correction de l'exercice 1 ▲

1. Soit P l'opérateur de projection dans le sous-espace U de dimension 1 généré par v. Alors $Q = I - P$ est l'opérateur de projection sur l'hyperplan $U^\perp$ orthogonal à U. On a déjà vu que $Pw = vv^T w \quad \forall w$, et donc $Qw = w - vv^T w$. On obtient
$P(H(v)w) = P(w_(2v^T w)v) = (v^T w)v - 2v^T wvv^T v = -(v^T w)v = -Pw$
$Q(H(v)w) = H(v)w - P(H(v)w) = w - 2vv^T w + v^T wv = w - v^T wv = Qw.$
La matrice $H(v)$ représente donc une symétrie par rapport à l'hyperplan $U^\perp$. On conclut que les vecteurs de $U^\perp$ sont invariants par $H(v)$.
$V(v)w = w \quad \forall w \in U^\perp, \quad \dim U^\perp = n-1 \Rightarrow \lambda = 1$ est valeur propre de $H(v)$ avec multiplicité $n-1$.
$H(v)v = -v \mp \lambda = -1$ est valeur propre de multiplicité 1. Donc
$$\det H(v) = \prod_{i=1}^{n} \lambda_i(H(v)) = -1$$

2. On sait qu'il exite des matrices de Householder $H_1, H_1, \ldots, H_{n-1}$ telles que $H_{n-1} \cdots H_1 A = A_n$ matrice triangulaire supérieure. Comme A est orthogonale on conclut que A_n est orthogonale. Mais une matrice triangulaire supérieure orthogonale est forcément diagonale $\Rightarrow A_n =$diag(± 1). On peut s'arranger pour que $(A_n)_{ii} > 0 \quad i = 1, \ldots, n-1$. Donc soit $A_n = I$ soit $A_n =$diag$(1, 1, \ldots, 1, -1) = H(e_n)$ et finalement la matrice orthogonale A s'écrit
$$A = H_1 \cdots H_{n-1} H(e_n)$$

Correction de l'exercice 2 ▲

1. Pour $k = 1, \cdots, n \quad a_k = \sum_{i=1}^{k} r_{ik} q_i$ avec $r_{ik} = q_i^T a_k$ par orthonormalité des q_i.
2. Découle immédiatement de la question précédente.
3. Algorithme de Gram-Schmidt :
Pour $k = 1, \cdots, n$ faire
$r_{ik} = q_i^T a_k \quad$ pour $i = 1, \cdots, k-1$
$z_k = a_k - \sum_{i=1}^{k-1} r_{ik} q_i$
$r_{kk} = (z_k^T z_k)^{1/2}$
$q_k = z_k / r_{kk}$
4. (a)
$$\sum_{i=k}^{n} q_i r_i^T = [q_k \cdots q_n] \begin{pmatrix} r_k^T \\ \vdots \\ r_n^T \end{pmatrix} = q_k \cdots q_n] \begin{pmatrix} 0 & \cdots & 0 & r_{kk} & \cdots & \cdots & r_{kn} \\ 0 & \cdots & 0 & 0 & r_{k+1,k+1} & \cdots & r_{k+1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & & & 0 & r_{nn} \end{pmatrix}$$
$$A^{(k)} e_k = z = [q_k \cdots q_n] \begin{pmatrix} r_{kk} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = r_{kk} q_k \Rightarrow r_{kk} = \|z\|_2, q_k = z / r_{kk}$$
(b)
$$q_k^T A^{(k)} = [q_k^T z, q_k^T B] = [1, 0, \cdots, 0] \begin{pmatrix} r_k^T \\ \vdots \\ r_n^T \end{pmatrix} = r_k^T$$
et donc
$$[r_{k,k+1}, \cdots, r_{kn}] = q_k^T B$$

(c)
$$[0,\cdots,0,A^{(k+1)}] = \sum_{i=k+1}^{n} q_i r_i^T = [0,\cdots,0,A^{(k)}] - q_k r_k^T = [0,\cdots,0,A^{(k)} - q_k(r_{kk},\cdots,r_{kn})]$$
$$[0,\cdots,0,z - q_k r_{kk}, B - q_k(r_{k,k+1},\cdots,r_{kn})] \Rightarrow A^{(k+1)} = B - q_k(r_{k,k+1},\cdots,r_{kn})$$

(d) Données : $A \in \mathbb{R}^{m\times n}$, $\mathrm{rank}(A) = n$
On calcule la factorisation $A = Q_1 R_1$, $Q_1 \in \mathbb{R}^{m\times n}$ orthonormale, $R_1 \in \mathbb{R}^{n\times n}$ triangulaire supérieure.
Le calcul de Q_1 se fait sur place.
Pour $k = 1,\cdots,n$
$r_{kk} = \left(\sum_{i=1}^m a_{ik}^2\right)^{1/2}$
pour $i = 1,\cdots,m$
$a_{ik} \leftarrow a_{ik}/r_{kk}$
pour $j = k+1,\cdots,n$
$r_{kj} \leftarrow \sum_{i=1}^m a_{ik}a_{ij}$
pour $i = 1,\cdots,m$
$a_{ij} \leftarrow a_{ij} - a_{ik}r_{kj}$

(e) complexité : mn^2 flops.

Correction de l'exercice 3 ▲

1. $G_{p,q}(c,s) = I + (c-1)e_p e_p^T + s e_q e_p^T - s e_q e_q^T + (c-1)e_p e_q^T$ avec e_i les vecteurs de la base canonique.
2. On montre que $e_i^T G^T G e_j = \delta_{ij} \quad \forall i,j = 1,\cdots,n$ et donc $G^T G = I$ ce qui permet de conclure que G est inversible d'inverse G^T et donc orthogonale.
3. $e_i^T GA = e_i^T A = a_i^T$ pour $i \neq p,q$
$e_p^T GA = ca_p^T - sa_q^T$, $e_q^T GA = sa_p^T + ca_q^T$, et donc G change seulement les lignes p et q
4. On pose $\alpha = a_{pj}$ et $\beta = a_{qj}$. On a donc à résoudre dans le premier cas le système
$$\begin{cases} c\alpha - s\beta = 0 \\ c^2 + s^2 = 1 \end{cases} \Leftrightarrow \begin{cases} c = \pm\beta/\sqrt{\alpha^2+\beta^2} \\ s = \pm\alpha/\sqrt{\alpha^2+\beta^2} \end{cases}$$
ce qui nous donne deux matrices G. Pour le deuxième cas et en procédant de la même façon on obtient
$$\begin{cases} c = \pm\alpha/\sqrt{\alpha^2+\beta^2} \\ s = \mp\beta/\sqrt{\alpha^2+\beta^2} \end{cases}$$

Correction de l'exercice 6 ▲

Méthode de Givens rapide

1. $MM^T = \mathrm{diag}(d_1,\cdots,d_m) = \Delta^2$ avec $\Delta = \mathrm{diag}(\sqrt{d_1},\cdots,\sqrt{d_m})$
$\Delta^{-1}MM^T\Delta^{-1} = (\Delta^{-1}M)(\Delta^{-1}M)^T = I \Rightarrow \Delta^{-1}M$ est une matrice orthogonale
$A = M^{-1}S = (M^{-1}Delta\Delta^{-1}S = (\Delta^{-1}M)^{-1}(\Delta^{-1}S) = (\Delta^{-1}M)^T(\Delta^{-1}S) = (M^T\Delta^{-1})(\Delta^{-1}S)$
Comme $\Delta^{-1}S$ est triangulaire supérieure ona $A = QR$ avec $Q = M^T\Delta^{-1}, R = \Delta^{-1}S$
2. (a)
$$M_1 x = \begin{pmatrix} \beta_1 x_1 + x_2 \\ x_1 + \alpha_1 x_2 \end{pmatrix}, \quad M_1 D M_1^T = \begin{pmatrix} d_2 + \beta_1^2 d_1 & d_1\beta_1 + d_2\alpha_1 \\ d_1\beta_1 + d_2\alpha_1 & d_1 + \alpha_1^2 d_2 \end{pmatrix}$$
Alors
– $x_1 + \alpha_1 x_2 = 0 \Leftrightarrow \alpha_1 = -x_1/x_2$
– $d_1\beta_1 + d_2(-x_1/x_2) = 0 \Leftrightarrow \beta_1 = -\alpha_1 d_2/d_1 = x_1 d_2/(x_2 d_1)$

Pour le choix précédent on veut déterminer γ_1 tel que
$x_2(1+\gamma_1) = \beta_1 x_1 + x_2 = x_2(\beta_1 x_1/x_2 + 1) \Rightarrow \gamma_1 = (d_2/d_1)(x_1/x_2)^2$ c'est-à-dire

$$\gamma_1 = -\alpha_1\beta_1$$

pour cette valeur on a
$d_2 + \beta_1^2 d_1 = d_2(1+\alpha_1^2 d_2/d_1) = d_2(1+\gamma_1)$ $d_1 + \alpha_1^2 d_2 = d_1(1+\alpha_1^2 d_2/d_1) = d_2(1+\gamma_1)$

(b) le même type de calcul nous donne

$$\beta_2 = -x_2/x_1, \quad \alpha_2 = -(d_1/d_2)\beta_2, \quad \gamma_2 = -\alpha_2\beta_2 = (d_1/d_2)(x_2/x_1)^2$$

(c) on remarque que $\gamma_1\gamma_2 = 1$ et donc soit $\gamma_1 \leq 1$, soit $\gamma_2 \leq 1$

3.
$$\begin{pmatrix} m_{pp} & m_{pq} \\ m_{qp} & m_{qq} \end{pmatrix} = \begin{pmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{pmatrix} \text{ ou } \begin{pmatrix} 1 & \alpha_2 \\ \beta_2 & 1 \end{pmatrix}$$

avec les α_i, β_i définis comme précédemment.

4. *algorithme*
$d_i = 1$ pour $i = 1, \cdots, m$
Pour $p = 1, \cdots, \min\{n, m-1\}$
    Pour $q = p+1, \cdots, m$
        si $a_{qp} \neq 0$ alors
            $\alpha = -a_{pp}/a_{qp}, \quad \beta = -\alpha d_q/d_p, \quad \gamma = -\alpha\beta$
            si $\gamma \leq 1$ alors
$$\begin{pmatrix} a_{pp} & \cdots & a_{pn} \\ a_{qp} & \cdots & a_{qn} \end{pmatrix} \leftarrow \begin{pmatrix} \beta & 1 \\ 1 & \alpha \end{pmatrix} \begin{pmatrix} a_{pp} & \cdots & a_{pn} \\ a_{qp} & \cdots & a_{qn} \end{pmatrix}$$
                échanger d_p et d_q
                    $d_p \leftarrow (1+\gamma d_p)$
                    $d_q \leftarrow (1+\gamma d_q)$
            sinon
                échanger α et β
                $\alpha = 1/\alpha$, $\beta = 1/\beta$, $\gamma = 1/\gamma$
$$\begin{pmatrix} a_{pp} & \cdots & a_{pn} \\ a_{qp} & \cdots & a_{qn} \end{pmatrix} \leftarrow \begin{pmatrix} 1 & \alpha \\ \beta & 1 \end{pmatrix} \begin{pmatrix} a_{pp} & \cdots & a_{pn} \\ a_{qp} & \cdots & a_{qn} \end{pmatrix}$$
                $d_p \leftarrow (1+\gamma d_p)$
                $d_q \leftarrow (1+\gamma d_q)$

le coût de cet algorithme est de $n^2(m-n/3)$ flops.

5. (a) on a $MA = R = \begin{pmatrix} S_1 \\ 0 \end{pmatrix}$ avec S_1 triangulaire supérieure et $MM^T = D = \text{diag}(d_1, \cdots, d_n)$. Donc la matrice $D^{-1/2}M$ est une matrice orthogonale

$$\|Ax-b\|_2^2 = \|D^{-1/2}MAx - D^{-1/2}Mb\|_2^2 = \left\| D^{-1/2}\left[\begin{pmatrix} S_1 \\ 0 \end{pmatrix} x - Mb\right]\right\|_2^2 =$$
$$= \left\| D^{-1/2}\left[\begin{pmatrix} S_1 \\ 0 \end{pmatrix} x - \begin{pmatrix} c \\ d \end{pmatrix}\right]\right\|_2^2 = \left\| D^{-1/2}\begin{pmatrix} S_1x - c \\ d \end{pmatrix}\right\|_2^2$$

La solution est obtenue en résolvant le système triangulaire supérieure $S_1x = c$ de taille $n \times n$.

(b) – mise à jour de b pour le calcul de Mb en même temps que la mise à jour de A
    pour $p = 1, \cdots, \min\{n, m-1\}$
        pour $q = p+1, \cdots, m$ faire
$$\begin{pmatrix} b_p \\ b_q \end{pmatrix} \leftarrow \begin{pmatrix} \beta & 1 \\ 1 & \alpha \end{pmatrix} \begin{pmatrix} b_p \\ b_q \end{pmatrix} \text{ ou}$$
$$\begin{pmatrix} b_p \\ b_q \end{pmatrix} \leftarrow \begin{pmatrix} 1 & \alpha \\ \beta & 1 \end{pmatrix} \begin{pmatrix} b_p \\ b_q \end{pmatrix}$$

– résolution du système triangulaire sup. $S_1 x = c$
$x_n \leftarrow b_n/a_{nn}$
Pour $i = n-1, \cdots, 1$ faire
$x_i \leftarrow (b_i - \sum_{j=i+1}^{n} a_{ij} x_j)$

6. Application numérique :

$$M = \frac{1}{24}\begin{pmatrix} 8 & 16 & 24 \\ 40 & 10 & -20 \\ 15 & -30 & 15 \end{pmatrix}, \quad D = \mathrm{diag}(14/9, 175/48, 75/32)$$

$$M[A,b] = \begin{pmatrix} 14/3 & 32/3 & 50/3 \\ 0 & 15/4 & 15/2 \\ 0 & 0 & 0 \end{pmatrix}, \quad x_{ls} = (-1,2)^T$$

7. on a

$$MD^{-2}M^T = \tilde{D} \Leftrightarrow (\tilde{D}^{-1/2}MD^{-1})(\tilde{D}^{-1/2}MD^{-1})^T = I$$

donc $(\tilde{D}^{-1/2}MD^{-1})$ est une matrice orthogonale et on obtient

$$\|D(Ax-b)\|_2 = \|\tilde{D}^{-1/2}MD^{-1}D(Ax-b)\|_2 = \|\tilde{D}^{-1/2}(MAx-Mb)\|_2 =$$

$$= \left|\tilde{D}^{-1/2}\begin{pmatrix} Sx-c \\ e \end{pmatrix}\right|_2$$

Donc le min est atteint pour $Sx = c$ avec $Mb = (C,e)^T$
La modification dans l'algorithme précédent consiste à initialiser la matrice diagonale D avec D^{-2} (au lieu de l'identité).
