Carte exponentielle de matrice par l'extension de l'algorithme de Jean-Marie Souriau, utilisable pour le tir géodésique et l'apprentissage machine pour les groupes de Lie

FREDERIC BARBARESCO[1]

[1] THALES Land & Air Systems
Voie Pierre-Gilles de Gennes, Limours 91470, France

[1]frederic.barbaresco@thalesgroup.com

Résumé - Jean-Marie Souriau a étendu en 1948 l'algorithme d'Urbain Jean Joseph Leverrier pour calculer le polynôme caractéristique d'une matrice. Cet algorithme de Souriau permet de calculer la carte exponentielle pour les groupes de Lie matriciels, ce qui est un défi en apprentissage machine sur les groupes de Lie. La propriété principale du schéma numérique de la carte exponentielle de Souriau est sa capacité à la haute parallélisation.

Abstract - Jean-Marie Souriau extended Urbain Jean Joseph Leverrier algorithm to compute characteristic polynomial of a matrix in 1948. This Souriau algorithm could be used to compute exponential map of a matrix that is a challenge in Lie Group Machine Learning. Main property of Souriau Exponential Map numerical scheme is its scalability by highly parallelization.

1 De l'algorithme de Leverrier à Souriau

En 1840, un algorithme permettant de calculer le polynôme caractéristique d'une matrice a été inventé par Urbain Jean Joseph Leverrier. Il a ensuite été redécouvert et amélioré en 1948 par Jean-Marie Souriau, mais n'a été publié qu'en français introduisant l'algorithme dans sa forme actuelle. Après Souriau, P. Horst, D.K. Faddejew et Sominski, J.S. Frame, U. Wegner et L. Csanky, ont été crédités d'avoir redécouvert cet algorithme. Dès 1955, l'algorithme de Souriau a été évalué par le National Bureau of Standards de Los Angeles, sous le parrainage du Wright Air Development Center, de l'US Air Force et de l'Office of Naval Research, et a été testé à l'Université de Californie par l'Office of Naval Research. Comme observé et illustré par Souriau, pour n = 10, son algorithme n'utilise que 8 000 additions et multiplications, contre 37 millions d'additions et 62 millions de multiplications pour l'approche classique. Pour les polynômes caractéristiques, les approches classiques sont basées sur les itérations de Krylov et l'élimination gaussienne. Mais l'algorithme de Krylov le plus efficace ne peut pas être parallélisé. En revanche, l'algorithme Souriau a été parallélisé avec succès par L. Csanky. L'algorithme de Souriau a une complexité $O(n^4)$ ou $O(n^{\omega+1})$ dans les calculs séquentiels et ne peut donc pas rivaliser avec l'algorithme basé sur Krylov. Mais l'algorithme Souriau est mieux adapté au calcul parallèle et l'algorithme de L. Csanky a prouvé que le calcul du polynôme caractéristique pouvait être résolu en parallèle en $log^2 n$ avec un nombre polynomial de processeurs. La parallélisation de l'algorithme de Souriau par Czansky a été améliorée par Preparata & Sarwate en utilisant un produit matriciel rapide et par Keller-Gehrig en utilisant une réduction matricielle. La réduction de complexité en $O(n^{\omega})$ est donné pour les matrices génériques, mais pour les matrices non génériques, seule la complexité en $O(n^{\omega} log n)$ peut être atteinte. L'algorithme de Faddejew-Leverrier a une complexité de même ordre de grandeur (essentiellement des multiplications nxn matricielles). C'est meilleur que la complexité que l'on pourrait obtenir dans un algorithme naïf basé sur le calcul du déterminant selon la formule de Leibniz: l'évaluation de $n!$ sommes conduit selon la formule de Stirling à une complexité temporelle de $\sqrt{O(n!)} = O\left(\sqrt{n}\left(\frac{n}{e}\right)^n\right)$, et le calcul au moyen de l'élimination gaussienne un ordre de grandeur en $O(n^3)$. Même si le calcul du déterminant pur est plus favorable, cependant, si l'on s'intéresse également aux coefficients du polynôme caractéristique, une complexité technique accrue lors de la mise en œuvre dans un programme informatique est nécessaire.

L'algorithme peut être efficacement parallélisé. Plus de détails peuvent être trouvés dans le travail original de Csanky ainsi que dans la vue d'ensemble des algorithmes pour le calcul parallèle du déterminant.

L'algorithme de Faddejew-Leverrier et l'élimination gaussienne présentent un désavantage, c'est la présence de divisions qui contrairement à la définition initiale du déterminant, ne nécessite pas de division et est donc applicable aux matrices dont les entrées sont des éléments d'un anneau commutatif. Dans ce cas, les algorithmes sans division, tels l'algorithme de Samuelson-Berkowitz, ont une complexité comparable.

2 Algorithme de Souriau de calcul du polynôme caractéristique d'une matrice

Jean-Marie Souriau a introduit son algorithme dans le contexte de son cours d'algèbre multi-linéaire, en considérant la *forme volume*.

Dans un espace vectoriel E de dimension *n*, on peut prouver que l'espace vectoriel de n-formes (*n-forme* comme un opérateur n-linéaire anti-symétrique avec des valeurs scalaires). Après avoir sélectionner un repère $(e_1, e_2, ..., e_n)$ de E, on peut définir une *n-forme* appelée "*forme volume* " avec:

$$vol(e_1)(e_2)...(e_n) = 1 \quad (1)$$

(le volume du parallélépipède généré par les vecteurs $(x_1, x_2, ..., x_n)$ est donné par :

$$|vol(x_1)(x_2)...(x_n)| \quad (2)$$

Souriau a appelé "espace jaugé", tout espace vectoriel E, de taille fini, pour lequel nous avons sélectionné une "jauge unité" définie par vol.

Si nous définissons un opérateur linéaire $A : E \to E$, considéré comme un "affineur" dans un espace jaugé, nous pouvons alors donner la définition de:

- ***Determinant de A,*** $\det(A)$ par:

$$\det(A) vol(v_1)(v_2)...(v_n) = vol(Av_1)(Av_2)...(Av_n) \quad (3)$$

- ***Opérateurr Lineaire Adjoint de A***, $adj(A)$, par:

$$vol(adj(A)v_1)(v_2)...(v_n) = vol(v_1)(Av_2)...(Av_n) \quad (4)$$

- ***Trace de A***, $tr(A)$, par:

$$\begin{aligned} tr(A) vol(v_1)(v_2)...(v_n) &= vol(Av_1)(v_2)...(v_n) \\ &+ vol(v_1)(Av_2)...(v_n) \\ &...+ vol(v_1)(v_2)...(Av_n) \end{aligned} \quad (5)$$

En utilisant la relation suivante déduite des équations précédentes :

$$\begin{aligned} vol(adj(A)Av_1)(v_2)...(v_n) &= vol(Av_1)(Av_2)...(Av_n) \\ &= \det(A) vol(v_1)(v_2)...(v_n) \end{aligned} \quad (6)$$

Si A est inversible, nous retrouvons les relations:

$$adj(A)A = \det(A) I \text{ et } A^{-1} = [\det(A)]^{-1} adj(A) \quad (7)$$

En utilisant ces formules, nous pouvons inverser $[\lambda I - A]$ en faisant l'hypothèse que $\det(\lambda I - A) \neq 0$.

Si on utilise la definition du determinant précédent:

$$\begin{aligned} &\det(\lambda I - A) vol(v_1)(v_2)...(v_n) = \\ &vol(\lambda v_1 - Av_1)(\lambda v_2 - Av_2)...(\lambda v_n - Av_n) \\ &= \lambda^n vol(v_1)(v_2)...(v_n) + ... \end{aligned} \quad (8)$$

où $\det(\lambda I - A)$ est le polynôme caractéristique de *A*, un polynôme en λ de degré *n*, avec l'équation:

$$\begin{aligned} &adj(\lambda I - A)[\lambda I - A] = \det(\lambda I - A) I \\ &\Leftrightarrow A.Q(\lambda) = \lambda Q(\lambda) - P(\lambda) I \end{aligned} \quad (9)$$

(Si λ est une valeur propre de A, les colonnes de non-zéros de $Q(\lambda)$ sont les vecteurs propres associés)

Nous pouvons alors observer que $adj(\lambda I - A)$ est un polynôme de degré *n-1*. Nous pouvons alors exprimer les polynômes comme suit:

$$P(\lambda) = \det(\lambda I - A) = \sum_{i=0}^{n} k_i \lambda^{n-i}$$

$$\text{et } Q(\lambda) = adj(\lambda I - A) = \sum_{i=0}^{n-1} \lambda^{n-i-1} B_i \quad (10)$$

avec $k_0 = 1$, $k_n = (-1)^n \det(A)$, $B_0 = I$ et

$$B_{n-1} = (-1)^{n-1} adj(A) \quad (11)$$

En développant l'équation suivante donnée par $adj(\lambda I - A)[\lambda I - A] = \det(\lambda I - A) I$, nous pouvons écrire:

$$\begin{aligned} &\sum_{i=0}^{n} k_i \lambda^{n-i} I = \sum_{i=0}^{n-1} \lambda^{n-i-1} B_i [\lambda I - A] \\ &= \lambda B_{n-1} + \sum_{i=1}^{n-1} \lambda^{n-i} [B_i - B_{i-1} A] - B_{n-1} A \end{aligned} \quad (12)$$

Par identification terme par terme, nous pouvons identifier l'expression des B_i :

$$\begin{cases} B_0 = I \\ B_i = B_{i-1} A + k_i I \quad , \; i = 1, ..., n-1 \\ B_{n-1} A + k_n I = 0 \end{cases} \quad (13)$$

Nous pouvons alors observer que $A^{-1} = -\frac{B_{n-1}}{k_n}$ et également retrouver le théorème de Cayley:

$$k_0 A^n + k_1 A^{n-1} + ... + k_{n-1} A + k_n I = 0 \quad (14)$$

Pour aller plus loin, il faut utiliser un résultat classique de l'analyse $\delta[\det(G)] = tr(adj(G)\delta G)$. Si on pose $G = (\lambda I - A)$ et $\delta = \frac{d}{d\lambda}$, on obtient finalement $tr(adj(\lambda I - A)) = \frac{d}{d\lambda} \det(\lambda I - A)$ fournissant:

$$\sum_{i=0}^{n-1} \lambda^{n-i-1} tr(B_i) = \frac{d}{d\lambda}\left(\sum_{i=0}^{n} k_i \lambda^{n-i}\right) = \sum_{i=0}^{n-1} (n-i) k_i \lambda^{n-i-1} \quad (15)$$

Nous pouvons alors déduire que:

$$tr(B_i) = (n-i) k_i \; , \; i = 0, ..., n-1$$

Comme $B_i = B_{i-1}A + k_i I$, $tr(B_i) = tr(B_{i-1}A) + n.k_i$,

alors $k_i = -\frac{tr(B_{i-1}A)}{i}$ (16)

Ce qui permet d'obtenir l'algorithme de Souriau:

$$k_0 = 1 \quad \text{et} \quad B_0 = I$$

$$\begin{cases} A_i = B_{i-1}A \quad , \quad k_i = -\frac{1}{i}tr(A_i), \; i = 1,...,n-1 \\ B_i = A_i + k_i I \quad \text{ou} \quad B_i = B_{i-1}A - \frac{1}{i}tr(B_{i-1}A)I \end{cases} \tag{17}$$

$$A_n = B_{n-1}A \quad \text{et} \quad k_n = -\frac{1}{n}tr(A_n)$$

$$\det(\lambda I - A) = \sum_{i=0}^{n} k_i \lambda^{n-1}$$

3 Algorithme de Souriau étendu pour la carte exponentielle

L'extension de l'algorithme de Souriau pour calculer la carte exponentielle d'une matrice est basée sur une analogie entre la propriété algébrique:

$$[\lambda I - A]^{-1} = \frac{Q(\lambda)}{P(\lambda)} \Leftrightarrow [\lambda I - A]Q(\lambda) = P(\lambda)I \tag{18}$$

et la propriété différentielle:

$$\left[I\frac{d}{dt} - A\right]Q\left(\frac{d}{dt}\right) = P\left(\frac{d}{dt}\right)I \tag{19}$$

Alors la fonction numérique γ vérifie $P\left(\frac{d}{dt}\right)\gamma = 0$ et:

$$\begin{aligned} P\left(\frac{d}{dt}\right)\gamma &= \sum_{i=0}^{n} k_i \gamma^{(n-i)} \\ P\left(\frac{d}{dt}\right)\gamma &= k_0\gamma^{(n)} + k_1\gamma^{(n-1)} + ... + k_{n-1}\gamma^{(1)} + k_n\gamma = 0 \end{aligned} \tag{20}$$

avec $\gamma^{(n)} = \frac{d^n\gamma(t)}{dt^n}$ n$^{\text{ième}}$ dérivée de la fonction γ, avec les conditions initiales:

$$\gamma(0) = \gamma^{(1)}(0) = ... = \gamma^{(n-2)} = 0 \quad \text{et} \quad \gamma^{(n-1)}(0) = 1 \tag{21}$$

Dans ce cas, la fonction matricielle $\Phi = Q\left(\frac{d}{dt}\right)\gamma$ est solution de l'équation différentielle $\frac{d\Phi}{dt} = A\Phi(t)$, avec la condition initiale $\Phi(0) = I$:

$$\begin{aligned} \Phi &= Q\left(\frac{d}{dt}\right)\gamma = \sum_{i=0}^{n-1}\gamma^{(n-i-1)}B_i \\ \Phi &= \gamma^{(n-1)}B_0 + \gamma^{(n-2)}B_1 + ... + \gamma B_{n-1} \end{aligned} \tag{22}$$

On peut alors observer que la carte exponentielle de la matrice tA est donnée par:

$$\begin{cases} e^{tA} = \sum_{i=0}^{n-1}\gamma^{(n-i-1)}B_i = \gamma^{(n-1)}B_0 + \gamma^{(n-2)}B_1 + ... + \gamma B_{n-1} \\ \text{avec } B_0 = I \text{ et } B_i = B_{i-1}A - \frac{tr(B_{i-1}A)}{i}I \end{cases} \tag{23}$$

où γ est solution de:

$$\begin{cases} \gamma \text{ telle que} \\ k_0\gamma^{(n)} + k_1\gamma^{(n-1)} + ... + k_{n-1}\gamma^{(1)} + k_n\gamma = 0 \\ \text{avec } k_i = -\frac{tr(B_{i-1}A)}{i} \\ \gamma(0) = ... = \gamma^{(n-2)} = 0 \quad \text{et} \quad \gamma^{(n-1)}(0) = 1 \end{cases} \tag{24}$$

La solution $\gamma(t)$ de l'équation différentielle ordinaire caractéristique est obtenue sur $[0,h]$ de l'intervalle spectrale d'intégration. Pour le reste, la fonction exponentielle $\Phi(t)$ est calculée par:

$$\Phi(ph) = (\Phi(h))^p \tag{25}$$

L'algorithme de Souriau étendu pour le calcul de la carte exponentielle d'une matrice A est donné par:

$$\begin{aligned} &1)\begin{cases} B_0 = I \text{ et } B_i = B_{i-1}A - \frac{tr(B_{i-1}A)}{i}I \\ k_0 = 1 \;,\; k_i = -\frac{tr(B_{i-1}A)}{i} \quad i = 1,...,n \end{cases} \\ &2)\begin{cases} \gamma \text{ intégré sur } [0,h] \text{ telle que} \\ k_0\gamma^{(n)} + k_1\gamma^{(n-1)} + ... + k_{n-1}\gamma^{(1)} + k_n\gamma = 0 \\ \text{avec } \gamma(0) = ... = \gamma^{(n-2)} = 0 \text{ et } \gamma^{(n-1)}(0) = 1 \end{cases} \end{aligned} \tag{26}$$

3) Calcul de $\Phi(t) = e^{tA} = \sum_{i=0}^{n-1}\gamma^{(n-i-1)}(t)B_i$ sur $[0,h]$

4) Extension du calcul sur $[0, ph]$

par $\Phi(pt) = (\Phi(t))^p$

5) $X(t) = \Phi(t)X_0$ avec $X_0 = X(0)$

Si on observe que $\ln(A) = \int_{-\infty}^{0}[sI - A]^{-1} - [sI - I]^{-1}ds$,

L'algorithme est utilisable pour $A^s = e^{s\ln(A)}$ avec $A^{1/2}$.

4 Exemple de carte exponentielle de matrice

On illustre pour les formes closes pour l'algèbre de Lie $\mathfrak{so}(3)$, du groupe de Lie $SO(3) = \{R / R^{-1} = R^T\}$:

$$\omega_\times = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \; \omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3 \tag{27}$$

$$\omega_\times = \omega_1 L_1 + \omega_2 L_2 + \omega_3 L_3 \in \mathfrak{so}(3)$$

L'exponentiel d'une matrice de $\mathfrak{so}(3)$ peut se decomposer avec l'algorithme de Souriau en:

$$e^{\omega_\times} = \gamma^{(2)} B_0 + \gamma^{(1)} B_1 + \gamma B_2 \quad (28)$$

Avec les paramètres B_i donnés par:

$$B_0 = I \text{ et } k_0 = 1 \quad (29)$$

$$B_1 = I.\omega_\times - \frac{Tr\left(I.\omega_\times\right)}{1} I = \omega_\times,\ k_1 = -\frac{Tr\left(I\omega_\times\right)}{1} = 0 \quad (30)$$

$$B_2 = B_1.\omega_\times - \frac{Tr\left(\omega_\times.\omega_\times\right)}{2} I = \omega_\times.\omega_\times + \left\|\omega\right\|^2 I$$
$$k_2 = -\frac{Tr\left(\omega_\times.\omega_\times\right)}{2} = \left\|\omega\right\|^2 \quad (31)$$

On peut observer que $B_2 = \omega_\times.\omega_\times + \left\|\omega\right\|^2 I = \omega \otimes \omega^T$ et $k_3 = 0$, ce qui fournit par la formule de Souriau:

$$e^{\omega_\times} = \gamma^{(2)} I + \gamma^{(1)} \omega_\times + \gamma \omega \otimes \omega^T \quad (32)$$

La fonction $\gamma(t)$ doit vérifier:

$$k_0\gamma^{(3)}(t) + k_1\gamma^{(2)}(t) + k_2\gamma^{(1)}(t) + k_3\gamma(t) = 0 \quad (33)$$

$$k_0 = 1,\ k_1 = 0,\ k_2 = \left\|\omega\right\|^2,\ k_3 = 0 \quad (34)$$

$$\gamma^{(3)}(t) + \left\|\omega\right\|^2 \gamma^{(1)}(t) = 0$$
$$\text{avec } \gamma^{(2)}(0) = 1, \gamma^{(1)}(0) = 0, \gamma(0) = 0 \quad (35)$$

On en déduit que la fonction $\gamma(t)$ est donnée par:

$$\gamma^{(1)}(t) = \frac{1}{\left\|\omega\right\|} \sin\left(\left\|\omega\right\| t\right), \quad \gamma(t) = \frac{1}{\left\|\omega\right\|^2}\left(1 - \cos\left(\left\|\omega\right\| t\right)\right)$$

$$e^{t.\omega_\times} = \cos\left(\left\|\omega\right\| t\right) I + \frac{1}{\left\|\omega\right\|} \sin\left(\left\|\omega\right\| t\right) \omega_\times + \frac{1 - \cos\left(\left\|\omega\right\| t\right)}{\left\|\omega\right\|^2} \omega \otimes \omega^T \quad (36)$$

Avec $\omega \otimes \omega^T = \omega_\times.\omega_\times + \left\|\omega\right\|^2 I$, on retrouve Rodrigue:

$$e^{t.\omega_\times} = I + \frac{1}{\left\|\omega\right\|} \sin\left(\left\|\omega\right\| t\right) \omega_\times + \frac{1 - \cos\left(\left\|\omega\right\| t\right)}{\left\|\omega\right\|^2} \omega_\times^2 \quad (37)$$

Un calcul identique s'applique pour $\mathfrak{se}(3)$ algèbre du groupe $SE(3) = \left\{ C / C = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, R \in SO(3), t \in \mathbb{R}^3 \right\}$:

$$\delta = \begin{pmatrix} \omega_\times & u \\ 0 & 0 \end{pmatrix},\ \delta = \begin{pmatrix} u & \omega \end{pmatrix} \in \mathfrak{se}(3) \text{ et } \begin{pmatrix} u & \omega \end{pmatrix}^T \in \mathbb{R}^6$$

$$\delta = u_1 G_1 + u_2 G_2 + u_3 G_3 + \omega_1 G_4 + \omega_2 G_5 + \omega_3 G_6$$

$$e^{\delta} = \exp\begin{pmatrix} \omega_\times & u \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} e^{\omega_\times} & Vu \\ 0 & 1 \end{pmatrix}, V = I + \frac{1}{2!}\omega_\times + \frac{1}{3!}\left(\omega_\times\right)^2 \ldots$$

Avec l'identité, $\left(\omega_\times\right)^3 = -\left(\omega^T \omega\right).\omega_\times = -\left\|\omega\right\|^2 \omega_\times$:

$$V = I + \sum_{i=0}^{\infty} \left[\frac{\omega_\times^{2i+1}}{(2i+2)!} + \frac{\omega_\times^{2i+2}}{(2i+3)!} \right] \quad (38)$$

$$V = I + \left(\sum_{i=0}^{\infty} \frac{(-1)^i \theta^{2i}}{(2i+2)!} \right) \omega_\times + \left(\sum_{i=0}^{\infty} \frac{(-1)^i \theta^{2i}}{(2i+3)!} \right) \omega_\times^2 \quad (39)$$

Pour retrouver finalement la formule connue:

$$V = I + \left(\frac{1 - \cos\left(\left\|\omega\right\|\right)}{\left\|\omega\right\|^2} \right) \omega_\times + \left(\frac{\left\|\omega\right\| - \sin\left(\left\|\omega\right\|\right)}{\left\|\omega\right\|^3} \right) \omega_\times^2 \quad (40)$$

L'algorithme de Souriau est utile en l'absence de formules closes.

5 Références

1. Souriau, J.-M..:Une méthode pour la décomposition spectrale et l'inversion des matrices. CRAS 227 (2), 1010–1011, Gauthier-Villars, (1948).
2. Souriau, J.-M.: Calcul Linéaire, Volume 1, "Euclide" Introduction aux études Scientifiques, Presses Universitaires de France, Paris, (1959)
3. Souriau, J.-M. ; Vallée, C. ; Réaud, K. ; Fortuné, D. : Méthode de Le Verrier–Souriau et équations différentielles linéaires, CRAS - Series IIB – Mechanics, 328 (10), 773-778, (2000)
4. Souriau, J.-M. : Grammaire de la Nature, communication privée, 8 July 2007
5. Réaud, K. ; Fortuné, D. ; Prudhorffne, S. ; Vallée, C., Méthode d'étude des vibrations d'un système mécanique non basée sur le calcul de ses modes propres, XVème Cong. Fr. de Méca., Nancy, (2001)
6. Champion-Réaud K., Méthode d'étude des vibrations d'un système mécanique non basée sur 1e calcul de ses modes propres Thèse SupAéro, (2002)
7. Réaud, K., Vallée, Cl. & Fortuné, D. Détermination des vecteurs propres d'un système vibratoire par exploitation du concept de matrice adjuguée, 6ème Col. national en calcul des structures, Giens. (2003)
8. Champion-Réaud, K. ; Vallée, C. ; Fortuné, D. Champion-Réaud, J.L., Extraction des pulsations et formes propres de la réponse d'un système vibratoire, 16ème Cong. Fr. de Méca., Nice, (2003)
9. Vallée, C ; Fortuné, D. ; Champion-Réaud, K., A General Solution of a Linear Dissipative Oscillatory System Avoiding Decomposition Into Eigenvectors, J. of Appl. Math. and Mech., 69, 837-843, (2005)
10. Moler, C. B. & van Loan, C. F.: Nineteen dubious ways to compute the exponential of a matrix, SIAM Rev. pp. 801–836., (1978)
11. Iserles A. and Zanna A.: Efficient Computation of the Matrix Exponential by Generalized Polar Decompositions, SIAM Journal on Numerical Analysis, Vol. 42, No. 5 : pp. 2218-2256, (2005)
12. Csanky, L.: Almost Parallel Matrix Inversion Algorithms. SIAM, 618-623, (1976)
13. Keller-Gehrig, W. : Fast algorithms for the characteristic polynomial. Theoretical computer science 36, 309–317, (1985).
14. Preparata, F.; Et Sarwate, D. : An improved parallel processor bound in fast matrix inversion. Information Processing Letters 7, 3, 148–150, (1978).