



Algorithme de Thomas

En algèbre linéaire appliquée à la résolution numérique d'équation, l'**algorithme de Thomas** (du nom de Llewellyn Thomas), est une forme simplifiée du pivot de Gauss qui peut être utilisée pour résoudre des systèmes d'équations tridiagonales. Un système tridiagonal pour n inconnues peut s'écrire

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i,$$

où $a_1 = 0$ et $c_n = 0$.

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & \ddots & \ddots & c_{n-1} & \\ 0 & & a_n & b_n & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}.$$

Pour de tels systèmes, la solution a une complexité algorithmique de $O(n)$ au lieu de $O(n^3)$ requis par élimination gaussienne. Un premier balayage élimine les coefficients a_i , puis une retro substitution (abrégée) produit la solution. Des exemples de telles matrices proviennent généralement de la discrétisation de l'équation de Poisson 1D et de l'interpolation par splines cubiques.

L'algorithme de Thomas n'est pas stable en général, mais il l'est dans plusieurs cas particuliers, notamment lorsque la matrice est diagonalement dominante (soit par lignes, soit par colonnes) ou symétrique définie positive^{1,2}. Pour une caractérisation plus précise de la stabilité de l'algorithme de Thomas, voir le théorème de Higham 9.12³. Si la stabilité est nécessaire dans le cas général, l'élimination gaussienne avec pivotement partiel (GEPP) est alors recommandée².

Méthode

Le balayage vers l'avant consiste en le calcul de nouveaux coefficients comme suit, où les nouveaux coefficients sont notés par des dérivées :

$$c'_i = \begin{cases} \frac{c_i}{b_i}, & i = 1, \\ \frac{c_i}{b_i - a_i c'_{i-1}}, & i = 2, 3, \dots, n-1 \end{cases}$$

et

$$d''_i = \begin{cases} \frac{d_i}{b_i}, & i = 1, \\ \frac{d_i - a_i d''_{i-1}}{b_i - a_i c'_{i-1}}, & i = 2, 3, \dots, n. \end{cases}$$

La solution est alors obtenue par substitution à rebours :

$$\begin{aligned} x_n &= d''_n, \\ x_i &= d''_i - c'_i x_{i+1}, \quad i = n-1, n-2, \dots, 1. \end{aligned}$$

La méthode ci-dessus ne modifie pas les vecteurs de coefficients d'origine, mais doit stocker les nouveaux coefficients. Si les vecteurs de coefficients doivent être modifiés, alors un algorithme avec moins de gestion de l'information est possible :

Pour $i = 2, 3, \dots, n$, on détermine

$$w = \frac{a_i}{b_{i-1}},$$

$$b_i := b_i - wc_{i-1},$$

$$d_i := d_i - wd_{i-1},$$

suivi du remplacement à rebours

$$x_n = \frac{d_n}{b_n},$$

$$x_i = \frac{d_i - c_i x_{i+1}}{b_i} \quad \text{pour } i = n-1, n-2, \dots, 1.$$

Calcul

L'algorithme matriciel tridiagonal est un cas particulier d'élimination gaussienne.

On suppose que les inconnues soient x_1, \dots, x_n , et que les équations à résoudre sont :

$$\begin{aligned} b_1 x_1 + c_1 x_2 &= d_1 \\ a_i x_{i-1} + b_i x_i + c_i x_{i+1} &= d_i, \quad i = 2, \dots, n-1 \\ a_n x_{n-1} + b_n x_n &= d_n. \end{aligned}$$

En modifiant la deuxième équation ($i = 2$) avec la première équation comme suit :

$$(\text{équation 2}) \cdot b_1 - (\text{équation 1}) \cdot a_2$$

ce qui donnerait :

$$(b_2 b_1 - c_1 a_2) x_2 + c_2 b_1 x_3 = d_2 b_1 - d_1 a_2.$$

On peut remarquer que x_1 a été éliminé de la deuxième équation. Utiliser une tactique similaire avec la deuxième équation **modifiée** sur la troisième équation donne :

$$(b_3(b_2b_1 - c_1a_2) - c_2b_1a_3)x_3 + c_3(b_2b_1 - c_1a_2)x_4 = d_3(b_2b_1 - c_1a_2) - (d_2b_1 - d_1a_2)a_3.$$

Cette fois x_2 a été éliminé. Si cette procédure est répétée pour l'ensemble des n lignes, alors chaque équation modifiée n'impliquerait qu'une seule inconnue, x_n . Cette équation peut être résolue puis utilisée pour résoudre l'équation $(n - 1)$, et ainsi de suite jusqu'à ce que toutes les inconnues soient obtenues.

On observe que les coefficients des équations modifiées deviennent de plus en plus compliqués s'ils sont exprimés de façon explicite. En examinant la procédure, les coefficients modifiés (notés par des tildes) peuvent plutôt être définis de manière récursive :

$$\tilde{a}_i = 0$$

$$\tilde{b}_1 = b_1$$

$$\tilde{b}_i = b_i \tilde{b}_{i-1} - \tilde{c}_{i-1} a_i$$

$$\tilde{c}_1 = c_1$$

$$\tilde{c}_i = c_i \tilde{b}_{i-1}$$

$$\tilde{d}_1 = d_1$$

$$\tilde{d}_i = d_i \tilde{b}_{i-1} - \tilde{d}_{i-1} a_i.$$

Pour accélérer encore la résolution, les coefficients \tilde{b}_i peuvent être divisés (s'il n'y a pas de risque de division par zéro), les coefficients modifiés les plus récents, ici notés par une dérivée, seront :

$$a'_i = 0$$

$$b'_i = 1$$

$$c'_1 = \frac{c_1}{b_1}$$

$$c'_i = \frac{c_i}{b_i - c'_{i-1} a_i}$$

$$d'_1 = \frac{d_1}{b_1}$$

$$d'_i = \frac{d_i - d'_{i-1} a_i}{b_i - c'_{i-1} a_i}.$$

Cela donne le système suivant :

$$\begin{aligned} x_i + c'_i x_{i+1} &= d'_i & ; \quad i = 1, \dots, n-1 \\ x_n &= d'_n & ; \quad i = n. \end{aligned}$$

La dernière équation ne comportant qu'une seule inconnue, en la résolvant on réduit l'avant-dernière équation à une inconnue, de sorte que cette substitution à rebours puisse être utilisée pour trouver toutes les inconnues :

$$\begin{aligned} x_n &= d'_n \\ x_i &= d'_i - c'_i x_{i+1} \quad ; \quad i = n-1, n-2, \dots, 1. \end{aligned}$$

Références

1. Pradip Niyogi, *Introduction to Computational Fluid Dynamics*, Pearson Education India, 2006 ([ISBN 978-81-7758-764-7](#)), p. 76
 2. Biswa Nath Datta, *Numerical Linear Algebra and Applications, Second Edition*, SIAM, 2010 ([ISBN 978-0-89871-765-5](#)), p. 162
 3. Nicholas J. Higham, *Accuracy and Stability of Numerical Algorithms: Second Edition*, SIAM, 2002 ([ISBN 978-0-89871-802-7](#)), p. 175
- W. H. Press, S. A. Teukolsky, W. T. Vetterling et B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, New York, Cambridge University Press, 2007 ([ISBN 978-0-521-88068-8](#)), « Section 2.4 »

Ce document provient de « https://fr.wikipedia.org/w/index.php?title=Algorithme_de_Thomas&oldid=216760321 ».