http://mmg.tversu.ru ISSN 2311-1275

Mathematical Modelling and Geometry

Volume 6, No 3, pp. 22 – 29 (2018) doi:10.26456/mmg/2018-633

Symbolic Algorithm for Solving SLAEs with Heptadiagonal Coefficient Matrices

Milena Veneva[1,a] **and Alexander Ayriyan**[1,b]

[1] Joint Institute for Nuclear Research, Laboratory of Information Technologies, Joliot-Curie 6, 141980 Dubna, Moscow region, Russia

e-mail: [a] milena.p.veneva@gmail.com, [b] ayriyan@jinr.ru

Received 27 November 2018. Published 19 December 2018.

Abstract. This paper presents a symbolic algorithm for solving band matrix systems of linear algebraic equations with heptadiagonal coefficient matrices. The algorithm is given in pseudocode. A theorem which gives the condition for the algorithm to be stable is formulated and proven.

Keywords: systems of linear algebraic equations, heptadiagonal matrix, symbolic algorithm, LU factorization

1. Introduction

Systems of linear algebraic equations (SLAEs) with heptadiagonal coefficient matrices may arise after many different scientific and engineering problems, as well as problems of the computational linear algebra where finding the solution of a SLAE is considered to be one of the most important problems. For instance, a semi-implicit formulation for the discretization of the transient terms of the system of partial differential equations (PDEs) which models a multiphase fluid flow in porous media yields to a heptadiagonal system of pressure equations for each time step (see [1]). On the other hand, in [2] the 3D problem, simulating the incompressible blood flow in arteries with a structured mesh domain leads to a heptadiagonal SLAE.

A whole branch of symbolic algorithms for solving systems of linear algebraic equations with different coefficient matrices exists in the literature. [3] considers a tridiagonal matrix and a symbolic version of the Thomas method [4] is formulated. The authors of [5] build an algorithm in the case of a general bordered tridiagonal SLAE, while in [6] the coefficient matrix taken into consideration is a general opposite-bordered tridiagonal one. A pentadiagonal coefficient matrix is of interest in [7], while a cyclic pentadiagonal coefficient matrix is considered in [8]. The latter algorithm can be applied to periodic tridiagonal and periodic pentadiagonal SLAE either by setting the corresponding matrix terms to be zero.

A performance analysis of effective methods (both numerical and symbolic) for solving band matrix SLAEs (with three and five diagonals) being implemented in `C++` and run on modern (as of 2018) computer systems is made by us in [9]. Different strategies (symbolic included) for solving band matrix SLAEs (with three and five diagonals) are explored by us in [11]. A performance analysis of effective symbolic algorithms for solving band matrix SLAEs with coefficient matrices with three, five and seven diagonals being implemented in both `C++` and `Python` and run on modern (as of 2018) computer systems is made by us in [10]. Note that the algorithm for solving a SLAE with a heptadiagonal coefficient matrix considered in [10] is the one that is going to be introduced in the next Section.

After obtaining the algorithm independently, it has been found in the article [12] where it is applied for cyclic heptadiagonal SLAEs. Thus, we do not claim out priority to this algorithm. However, the novelties of this work are as follows: pure heptadiagonality, proved necessary requirements, classical Thomas expressions i. e. the algorithm's formalism follows the form of expressions that are usually used in the Thomas algorithm for tridiagonal SLAE [4], that is, the solution is searched in the form: $y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}$.

2. The Algorithm

Let us consider a SLAE $Ax = y$, where A is a $N \times N$ heptadiagonal matrix, $A = \text{heptadiag}(\mathbf{c}^*, \mathbf{b}^*, \mathbf{a}^*, \mathbf{d}, \mathbf{a}, \mathbf{b}, \mathbf{c})$, x and y are vectors of length N:

$$\begin{pmatrix} d_0 & a_0 & b_0 & c_0 & 0 & \dots & \dots & \dots & \dots & 0 \\ a_1^* & d_1 & a_1 & b_1 & c_1 & 0 & \dots & \dots & \dots & 0 \\ b_2^\star & a_2^\star & d_2 & a_2 & b_2 & c_2 & 0 & \dots & \dots & 0 \\ c_3^\star & b_3^\star & a_3^\star & d_3 & a_3 & b_3 & c_3 & 0 & \dots & 0 \\ 0 & c_4^\star & b_4^\star & a_4^\star & d_4 & a_4 & b_4 & c_4 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & c_{N-4}^\star & b_{N-4}^\star & a_{N-4}^\star & d_{N-4} & a_{N-4} & b_{N-4} & c_{N-4} \\ 0 & \dots & \dots & 0 & c_{N-3}^\star & b_{N-3}^\star & a_{N-3}^\star & d_{N-3} & a_{N-3} & b_{N-3} \\ 0 & \dots & \dots & \dots & 0 & c_{N-2}^\star & b_{N-2}^\star & a_{N-2}^\star & d_{N-2} & a_{N-2} \\ 0 & \dots & \dots & \dots & \dots & 0 & c_{N-1}^\star & b_{N-1}^\star & a_{N-1}^\star & d_{N-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{N-4} \\ x_{N-3} \\ x_{N-2} \\ x_{N-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{N-4} \\ y_{N-3} \\ y_{N-2} \\ y_{N-1} \end{pmatrix}.$$

A symbolic algorithm for solving SLAEs with a heptadiagonal coefficient matrix is considered. It is based on LU decomposition in which the system $Ax = y$ is rewritten as $LUx = y$, where L and U are a lower triangular and an upper triangular matrices, respectively. The algorithm consists of two steps – the LU decomposition together with the downwards sweep $Lz = y$ happen during the first step, leading us from $Ax = y$ to $Ux = z$, while the upwards sweep (solving $Ux = z$ for x) is done during the second step.

Remark: seems that the expression for g_{i-2} is missing in the `for` loop on p. 435 of [12]. Also, the expression for k_1 on p. 435 of the same paper is not used anywhere, so it is probably a leftover from a previous algorithm.

Now we shall formulate a symbolic algorithm for solving such a SLAE.

Algorithm 1. Symbolic algorithm for solving a SLAE $Ax = y$.

Input: $\mathbf{c}^*, \mathbf{b}^*, \mathbf{a}^*, \mathbf{d}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{y}, \varepsilon$
Output: $\mathbf{x}$
if $det(A) == 0$ **then**
    Exit.
end if
bool flag = False
$\mu_0 := d_0$ ▷ Step 1.(0)
if $|\mu_0| < \varepsilon$ **then**
    $\mu_0 :=$ symb; flag = True
end if
$\alpha_0 := \dfrac{a_0}{\mu_0}; \quad \beta_0 := \dfrac{b_0}{\mu_0}; \quad \gamma_0 := \dfrac{c_0}{\mu_0}; \quad \delta_0 := 0.0$
$\delta_1 := a_1^*; \quad \mu_1 := d_1 - \alpha_0\delta_1$ ▷ (1)
if !flag **then**

if $|\mu_1| < \varepsilon$ **then**
$\mu_1 :=$ symb; flag = True
end if
end if
$\alpha_1 := \dfrac{a_1 - \beta_0\delta_1}{\mu_1}; \quad \beta_1 := \dfrac{b_1 - \gamma_0\delta_1}{\mu_1}; \quad \gamma_1 := \dfrac{c_1}{\mu_1}$
$\delta_2 := a_2^* - \alpha_0 b_2^*; \quad \mu_2 := d_2 - \alpha_1\delta_2 - \beta_0 b_2^*$ ▷ (2)
if !flag **then**
if $|\mu_2| < \varepsilon$ **then**
$\mu_2 :=$ symb; flag = True
end if
end if
$\alpha_2 := \dfrac{a_2 - \beta_1\delta_2 - \gamma_0 b_2^*}{\mu_2}; \quad \beta_2 := \dfrac{b_2 - \gamma_1\delta_2}{\mu_2}; \quad \gamma_2 := \dfrac{c_2}{\mu_2}$
$z_0 := \dfrac{y_0}{\mu_0}; \quad z_1 := \dfrac{y_1 - \delta_1 z_0}{\mu_1}; \quad z_2 := \dfrac{y_2 - \delta_2 z_1 - b_2^* z_0}{\mu_2}$
$\xi_0 := 0.0; \quad \xi_1 := 0.0; \quad \xi_2 := 0.0$
for $i = 3, \ldots N - 4$ **do** ▷ (i)
$\delta_i := a_i^* - \alpha_{i-2} b_i^* - c_i^*(\beta_{i-3} - \alpha_{i-3}\alpha_{i-2})$
$\xi_i := b_i^* - \alpha_{i-3} c_i^*$
$\mu_i := d_i - \alpha_{i-1}\delta_i - \beta_{i-2}\xi_i - \gamma_{i-3} c_i^*$
if !flag **then**
if $|\mu_i| < \varepsilon$ **then**
$\mu_i :=$ symb; flag = True
end if
end if
$\alpha_i := \dfrac{a_i - \beta_{i-1}\delta_i - \gamma_{i-2}\xi_i}{\mu_i}; \quad \beta_i := \dfrac{b_i - \gamma_{i-1}\delta_i}{\mu_i}; \quad \gamma_i := \dfrac{c_i}{\mu_i}$
$z_i := \dfrac{y_i - \delta_i z_{i-1} - \xi_i z_{i-2} - c_i^* z_{i-3}}{\mu_i}$
end for
$\delta_{N-3} := a_{N-3}^* - \alpha_{N-5} b_{N-3}^* - c_{N-3}^*(\beta_{N-6} - \alpha_{N-6}\alpha_{N-5})$ ▷ $(N-3)$
$\xi_{N-3} := b_{N-3}^* - \alpha_{N-6} c_{N-3}^*$
$\mu_{N-3} := d_{N-3} - \alpha_{N-4}\delta_{N-3} - \beta_{N-5}\xi_{N-3} - \gamma_{N-6} c_{N-3}^*$
if !flag **then**
if $|\mu_{N-3}| < \varepsilon$ **then**
$\mu_{N-3} :=$ symb; flag = True
end if
end if
$\alpha_{N-3} := \dfrac{a_{N-3} - \beta_{N-4}\delta_{N-3} - \gamma_{N-5}\xi_{N-3}}{\mu_{N-3}}$
$\beta_{N-3} := \dfrac{b_{N-3} - \gamma_{N-4}\delta_{N-3}}{\mu_{N-3}}$
$z_{N-3} := \dfrac{y_{N-3} - \delta_{N-3} z_{N-4} - \xi_{N-3} z_{N-5} - c_{N-3}^* z_{N-6}}{\mu_{N-3}}$

$\delta_{N-2} := a^*_{N-2} - \alpha_{N-4} b^*_{N-2} - c^*_{N-2}(\beta_{N-5} - \alpha_{N-5}\alpha_{N-4})$ ▷ $(N-2)$
$\xi_{N-2} := b^*_{N-2} - \alpha_{N-5} c^*_{N-2}$
$\mu_{N-2} := d_{N-2} - \alpha_{N-3}\delta_{N-2} - \beta_{N-4}\xi_{N-2} - \gamma_{N-5} c^*_{N-2}$
if !flag **then**
  if $|\mu_{N-2}| < \varepsilon$ **then**
    $\mu_{N-2} :=$ symb; flag = True
  end if
end if
$\alpha_{N-2} := \dfrac{a_{N-2} - \beta_{N-3}\delta_{N-2} - \gamma_{N-4}\xi_{N-2}}{\mu_{N-2}}$
$z_{N-2} := \dfrac{y_{N-2} - \delta_{N-2} z_{N-3} - \xi_{N-2} z_{N-4} - c^*_{N-2} z_{N-5}}{\mu_{N-2}}$
$\delta_{N-1} := a^*_{N-1} - \alpha_{N-3} b^*_{N-1} - c^*_{N-1}(\beta_{N-4} - \alpha_{N-4}\alpha_{N-3})$ ▷ $(N-1)$
$\xi_{N-1} := b^*_{N-1} - \alpha_{N-4} c^*_{N-1}$
$\mu_{N-1} := d_{N-1} - \alpha_{N-2}\delta_{N-1} - \beta_{N-3}\xi_{N-1} - \gamma_{N-4} c^*_{N-1}$
if !flag **then**
  if $|\mu_{N-1}| < \varepsilon$ **then**
    $\mu_{N-1} :=$ symb; flag = True
  end if
end if
$z_{N-1} := \dfrac{y_{N-1} - \delta_{N-1} z_{N-2} - \xi_{N-1} z_{N-3} - c^*_{N-1} z_{N-4}}{\mu_{N-1}}$
$x_{N-1} := z_{N-1}$ ▷ Step 2. Solution
$x_{N-2} := z_{N-2} - \alpha_{N-2} x_{N-1}$
$x_{N-3} := z_{N-3} - \alpha_{N-3} x_{N-2} - \beta_{N-3} x_{N-1}$
for $j = \overline{N-4, \ldots 0}$ **do**
  $x_j := z_j - \alpha_j x_{j+1} - \beta_j x_{j+2} - \gamma_j x_{x+3}$
end for
Cancel the common factors in the numerators and denominators of $\mathbf{x}$, making them coprime. Substitute symb := 0 in $\mathbf{x}$ and simplify.

Remark: If any μ_i expression has been evaluated to be zero or numerically zero, then it is assigned to be a symbolic variable. We cannot compare any of the next μ expressions with ε, because any further μ is going to be a symbolic expression. To that reason, we use a boolean flag which tells us if any previous μ is a symbolic expression. In that case, comparison with ε is not conducted as being not needed.

3. Stability of the Algorithm

Some observations on the stability of the proposed algorithm can be made. Firstly, assigning $\mu_i, i = \overline{0, N-1}$ to be equal to a symbolic variable in case it is zero or numerically zero, ensures correctness of the formulae for computing the solution of the considered SLAE. This action does not add any additional requirements to the coefficient matrix, except:

Theorem 1. *The only requirement to the coefficient matrix so as the algorithm to be stable is nonsingularity.*

Proof. As a direct consequence of the transformations done so as the matrix A to be factorized and then the downwards sweep to be conducted, it follows that the determinant of the matrix A in the terms of the introduced notation is:

$$\det(A) = \prod_{i=0}^{N-1} \mu_i|_{\text{symb}=0}. \tag{1}$$

(This formula could be used so as the nonsingularity of the coefficient matrix to be checked.) If μ_i for any i is assigned to be equal to a symbolic variable, then it is going to appear in both the numerator and the denominator of the expression for the determinant and so it can be cancelled:

$$\begin{aligned}
\det(A) &= \mu_0\,\mu_1\,\mu_2\ldots\mu_{N-2}\,\mu_{N-1} = \\
&= M_0\,\frac{M_1}{\mu_0}\,\frac{M_2}{\mu_0\,\mu_1}\ldots\frac{M_{N-2}}{\mu_0\,\mu_1\ldots\mu_{N-3}}\,\frac{M_{N-1}}{\mu_0\,\mu_1\ldots\mu_{N-2}} = \\
&= \frac{\prod_{i=0}^{N-1} M_i}{\mu_0^{N-1}\,\mu_1^{N-2}\,\mu_2^{N-3}\ldots\mu_{N-3}^{2}\,\mu_{N-2}^{1}} = \\
&= \frac{\prod_{i=0}^{N-1} M_i}{M_0^{N-1}\,\frac{M_1^{N-2}}{\mu_0^{N-2}}\,\frac{M_2^{N-3}}{\mu_0^{N-3}\,\mu_1^{N-3}}\ldots\frac{M_{N-3}^{2}}{\mu_0^2\,\mu_1^2\ldots\mu_{N-4}^2}\,\frac{M_{N-2}^{1}}{\mu_0^1\,\mu_1^1\ldots\mu_{N-3}^1}} = \\
&= \frac{\prod_{i=0}^{N-1} M_i}{M_0^{N-1}\,\frac{M_1^{N-2}}{\mu_0^{N-2}}\,\frac{M_2^{N-3}}{\mu_0^{N-3}\left(\frac{M_1}{\mu_0}\right)^{N-3}}\ldots\frac{M_{N-3}^{2}}{\mu_0^2\left(\frac{M_1}{\mu_0}\right)^2\ldots\left(\frac{M_{N-4}}{\mu_{N-3}}\right)^2}\,\frac{M_{N-2}^{1}}{\mu_0^1\left(\frac{M_1}{\mu_0}\right)^1\ldots\left(\frac{M_{N-3}}{\mu_{N-4}}\right)^1}} = \\
&= \frac{\prod_{i=0}^{N-1} M_i}{\prod_{i=0}^{N-2} M_i} = M_{N-1}
\end{aligned} \tag{2}$$

where M_i is the i-th leading principal minor, and $\mu_0 = M_0$. This means that the only constraint on the coefficient matrix is $M_{N-1} \neq 0$. □

The requirement on the coefficient matrix to be nonsingular is not limiting at all since this is a standard requirement so as the SLAE to have one solution only.

4. Conclusions

A symbolic algorithm for solving band matrix SLAEs with heptadiagonal coefficient matrices was presented in pseudocode. Some notes on the stability of the algorithm were made.

Acknowledgements. The work is partially supported by the Russian Foundation for Basic Research under project #18-51-18005.

References

[1] Kim J.G., Park H.W. *Advanced Simulation Technique for Modeling Multiphase Fluid Flow in Porous Media.* In: Laganá A., Gavrilova M.L., Kumar V., Mun Y., Tan C.J.K., Gervasi O. et al. Computational Science and Its Applications – ICCSA 2004. ICCSA 2004. Lecture Notes in Computer Science, 2004, **3044**. Springer, Berlin, Heidelberg, pp. 1 – 9, doi:10.1007/978-3-540-24709-8-1

[2] Duran A., Celebi M.S., Piskin S. et al. *Scalability of OpenFOAM for biomedical flow simulations.* The Journal of Supercomputing, 2015, **71**, pp. 938 – 951, doi:10.1007/s11227-014-1344-1

[3] El-Mikkawy, M. *A Generalized symbolic Thomas algorithm.* Applied Mathematics, 2012, **3**, 4, pp. 342 – 345, doi:10.4236/am.2012.34052

[4] Higham N. J. *Accuracy and Stability of Numerical Algorithms.* SIAM, 2nd edn, 2002, pp. 174 – 176

[5] Karawia A. A., Rizvi Q. M. *On solving a general bordered tridiagonal linear system.* International Journal of Mathematical Sciences, 2013, **33**, 2

[6] Atlan F., El-Mikkawy M. *A new symbolic algorithm for solving general opposite-bordered tridiagonal linear systems.* American Journal of Computational Mathematics, 2015, **5**, pp. 258 – 266, doi:10.4236/ajcm.2015.53023

[7] Askar S. S., Karawia A. A. *On solving pentadiagonal linear systems via transformations.* Mathematical Problems in Engineering. Hindawi Publishing Corporation, 2015, **2015**, 9, doi:10.1155/2015/232456

[8] Jia J.-T., Jiang Y.-L. *Symbolic algorithm for solving cyclic penta-diagonal linear systems.* Numerical Algorithms, 2012, **63**, 2, pp. 357 – 367, doi:10.1007/s11075-012-9626-2

[9] Veneva M., Ayriyan A. *Performance Analysis of Effective Methods for Solving Band Matrix SLAEs after Parabolic Nonlinear PDEs.* Advanced Computing in Industrial Mathematics, Revised Selected Papers of the 12th Annual Meeting of the Bulgarian Section of SIAM, December 20-22, 2017, Sofia, Bulgaria, Studies in Computational Intelligence, vol. 793, Springer International Publishing (X), pp. 407–419, arXiv:1804.09666

[10] Veneva M., and Ayriyan A. *Performance analysis of effective symbolic methods for solving band matrix SLAEs.* Submitted to European Physics Journal – Web of Conferences (EPJ-WoC)

[11] Veneva M., Ayriyan A. *Effective Methods for Solving Band SLAEs after Parabolic Nonlinear PDEs.* AYSS-2017, European Physics Journal – Web of Conferences (EPJ-WoC). 2018, **177**, 07004, arXiv:1710.00428

[12] Karawia A. A. *A new algorithm for general cyclic heptadiagonal linear systems using Sherman-Morrisor-Woodbury formula.* ARS Combinatoria, 2013, **108**, pp. 431–443, arXiv:1011.4580