Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (1) 46–56
http://journals.rudn.ru/miph

Research article

UDC 519.872, 519.217
PACS 07.05.Tp, 02.60.Pn, 02.70.Bf
DOI: 10.22363/2658-4670-2025-33-1-46-56
EDN: BLFUDE

Symbolic algorithm for solving SLAEs with multi-diagonal coefficient matrices

Milena Veneva[1, 2]

[1] Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russian Federation

[2] RIKEN Center for Computational Science, R-CCS, 7-1-26 Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan

(received: November 8, 2024; revised: November 30, 2024; accepted: December 12, 2024)

Abstract. Systems of linear algebraic equations with multi-diagonal 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 such a system of linear algebraic equations is considered to be one of the most important problems. This paper presents a generalised symbolic algorithm for solving systems of linear algebraic equations with multi-diagonal coefficient matrices. The algorithm is given in a pseudocode. A theorem which gives the condition for correctness of the algorithm is formulated and proven. Formula for the complexity of the multi-diagonal numerical algorithm is obtained.

Key words and phrases: numerical analysis, computational methods for sparse matrices, numerical mathematical programming methods, complexity of numerical algorithms

For citation: Veneva, M. Symbolic algorithm for solving SLAEs with multi-diagonal coefficient matrices. *Discrete and Continuous Models and Applied Computational Science* **33** (1), 46–56. doi: 10.22363/2658-4670-2025-33-1-46-56. edn: BLFUDE (2025).

1. Introduction

Systems of linear algebraic equations (SLAEs) with multi-diagonal 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, the resultant SLAE after discretization of partial differential equations (PDEs), using finite difference methods (FDM) or finite element methods (FEM) has a banded coefficient matrix. The methods for solving such SLAEs known in the literature usually require the matrix to possess special characteristics so as the method to be numerically correct and stable, e. g. diagonal dominance, positive definiteness, etc. Another possible approach which ensures numerically correct formulae without adding special additional requirements or using pivoting is the symbolic algorithms.

By definition, a band SLAE is a SLAE with band coefficient matrix. The lower band width p is the number of sub-diagonals, the upper band width q is the number of super-diagonals, and the band width of the matrix is defined as $p + q + 1$ (we should add 1 because of the main diagonal), that is, the total number of non-zero diagonals in the matrix [1]. Here, we are going to focus on SLAEs with matrices for which $p = q = M$. The author of [2] presents a generalised numerical algorithm for solving multi-diagonal SLAEs with pivoting (implemented in `Fortran`) in the case $p \neq q$, and has applied it for solving boundary value problems discretized by finite difference approximations.

A whole branch of symbolic algorithms for solving systems of linear algebraic equations with different coefficient matrices exists in the literature. For instance, in [3] the author considers a tridiagonal matrix and a symbolic version of the Thomas method [4–7] is formulated. The authors of [8] build an algorithm in the case of a general bordered tridiagonal SLAE, while in [9] the coefficient matrix taken into consideration is a general opposite-bordered tridiagonal one.

A pentadiagonal coefficient matrix is of interest in [10], while a cyclic pentadiagonal coefficient matrix is considered in [11]. The latter algorithm can be applied to periodic tridiagonal and periodic pentadiagonal SLAE either by setting the corresponding matrix terms to be zero.

In [12] a symbolic method for the case of a cyclic heptadiagonal SLAEs is presented.

What is common for all these symbolic algorithms, is that they are implemented using Computer Algebra Systems (CASs) such as `Maple` [13], `Mathematica` [14], and `Matlab` [15].

Finally, [16] presents a symbolic method for the case of a pure heptadiagonal SLAE.

A performance analysis of symbolic methods (and numerical as well) 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 in [17]. Different strategies (symbolic included) for solving band matrix SLAEs (with three and five diagonals) are explored in [18]. 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 in [19].

Having in mind all these introductory notes, it is clear that a generalised multi-diagonal symbolic algorithm is the novelty that addresses the need of a direct method which solves multi-diagonal systems of linear algebraic equations without putting any requirements for the characteristics of the coefficient matrix. Thus, the aim of this paper, which is a logical continuation of [16–19], is to present such a generalised symbolic algorithm for solving SLAEs with multi-diagonal coefficient matrices. The symbolic algorithms investigated in [16–19] are actually particular cases of the generalised multi-diagonal symbolic method when $p = q = M = 1, 2$, and 3.

The layout of the paper is as follows: in the next section, we outline the multi-diagonal numerical algorithm, and introduce the multi-diagonal symbolic algorithm in pseudocode. Afterwards, we make some correctness remarks for the symbolic method, and present a generalised formula for the complexity of the multi-diagonal numerical algorithm. Finally, some conclusions are drawn.

The novelties of this work are as follows: suggested multi-diagonal symbolic algorithm for solving SLAEs, formulation and proof of a correctness theorem, and an additionally obtained formula for the complexity of the multi-diagonal numerical method.

2. Multi-diagonal symbolic algorithm

Let us consider a SLAE $Ax = y$, where $A = \mathrm{diag}(\mathbf{b^0}, \mathbf{b^1}, \mathbf{b^2}, \dots, \mathbf{b^M}, \mathbf{b^{M+1}}, \mathbf{b^{M+2}}, \mathbf{b^{M+3}}, \dots, \mathbf{b^{2\times M}})$, A is a real $N \times N$ multi-diagonal matrix with M sub-diagonals, and M super-diagonals, and $2 \times M + 1 < N$, that is, the number of diagonals should be smaller than the number of equations within the SLAE; x

and y are real column vectors with N elements:

$$\begin{bmatrix} b_0^M & b_0^{M+1} & b_0^{M+2} & b_0^{M+3} & \cdots & b_0^{2M} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ b_1^{M-1} & b_1^M & b_1^{M+1} & b_1^{M+2} & \cdots & b_1^{2M-1} & b_1^{2M} & 0 & \cdots & \cdots & \cdots & 0 \\ b_2^{M-2} & b_2^{M-1} & b_2^M & b_2^{M+1} & \cdots & b_2^{2M-2} & b_2^{2M-1} & b_2^{2M} & 0 & \cdots & \cdots & 0 \\ b_3^{M-3} & b_3^{M-2} & b_3^{M-1} & b_3^M & \cdots & b_3^{2M-3} & b_3^{2M-2} & b_3^{2M-1} & b_3^{2M} & 0 & \cdots & 0 \\ b_4^{M-4} & b_4^{M-3} & b_4^{M-2} & b_4^{M-1} & \cdots & b_4^{2M-4} & b_4^{2M-3} & b_4^{2M-2} & b_4^{2M-1} & b_4^{2M} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & b_{N-4}^0 & b_{N-4}^1 & \cdots & \cdots & b_{N-4}^{M-1} & b_{N-4}^M & b_{N-4}^{M+1} & b_{N-4}^{M+2} & b_{N-4}^{M+3} \\ 0 & \cdots & \cdots & 0 & b_{N-3}^0 & b_{N-3}^1 & \cdots & \cdots & b_{N-3}^{M-1} & b_{N-3}^M & b_{N-3}^{M+1} & b_{N-3}^{M+2} \\ 0 & \cdots & \cdots & \cdots & 0 & b_{N-2}^0 & b_{N-2}^1 & \cdots & \cdots & b_{N-2}^{M-1} & b_{N-2}^M & b_{N-2}^{M+1} \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & b_{N-1}^0 & b_{N-1}^1 & \cdots & \cdots & b_{N-1}^{M-1} & b_{N-1}^M \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{N-4} \\ x_{N-3} \\ x_{N-2} \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{N-4} \\ y_{N-3} \\ y_{N-2} \\ y_{N-1} \end{bmatrix}.$$

The multi-diagonal numerical solver which we are going to formulate below is a generalization of the Thomas method for multi-diagonal SLAEs. The algorithm is based on LU decomposition, and requires forward reduction for reducing the initial matrix into a lower triangular one:

$$\mu_0 = b_0^M$$
$$\alpha_0^{M+1} = \frac{b_0^{M+1}}{\mu_0}$$
$$\alpha_0^{M+2} = \frac{b_0^{M+2}}{\mu_0}$$
$$\ldots$$
$$\alpha_0^{2M} = \frac{b_0^{2M}}{\mu_0}$$
$$z_0 = \frac{y_0}{\mu_0}$$

$$\alpha_1^j = 0, \quad i = 1, 2 \ldots, M-1$$
$$\alpha_1^M = b_1^{M-1}$$
$$\mu_1 = b_1^M - \alpha_0^{M+1} \times \alpha_1^M$$
$$\alpha_1^{M+1} = \frac{b_1^{M+1} - \alpha_0^{M+2} \times \alpha_1^M}{\mu_1}$$
$$\alpha_1^{M+2} = \frac{b_1^{M+2} - \alpha_0^{M+3} \times \alpha_1^M}{\mu_1}$$
$$\ldots$$
$$\alpha_1^{2M} = \frac{b_1^{2M}}{\mu_1}$$
$$z_1 = \frac{y_1 - z_0 \times \alpha_1^M}{\mu_1}$$

For $i = 2, 3, \ldots, M-1$:

counter $= M - i$

$$\alpha_i^{k-\text{counter}} = b_i^{k-1}, \quad k = M, M-1, \ldots, 1, \quad k - \text{counter} \geq 1$$
$$\alpha_i^{k-\text{counter}} = \alpha_i^{k-\text{counter}} - \alpha_0^{M+k-\text{counter}-1} \times \alpha_i^1 - \alpha_1^{M+k-\text{counter}-2} \times \alpha_i^2 - \ldots$$
$$- \alpha_{k-\text{counter}-2}^{M+1} \times \alpha_i^{k-\text{counter}-1}, \quad k = 2, 3, \ldots, M$$
$$\mu_i = b_i^M - \alpha_0^{M+i} \times \alpha_i^1 - \alpha_1^{M+i-1} \times \alpha_i^2 - \cdots - \alpha_{i-1}^{M+1} \times \alpha_i^i$$
$$\alpha_i^{M+1} = \frac{b_i^{M+1} - \alpha_0^{M+i+1} \times \alpha_i^1 - \alpha_1^{M+i} \times \alpha_i^2 - \cdots - \alpha_{i-1}^{M+2} \times \alpha_i^i}{\mu_i}$$

num_sub $= min(i, M-2)$ number of subtractions for α_i^{M+2}

$$\alpha_i^{M+2} = \frac{1}{\mu_i} \left(b_i^{M+2} - \alpha_{i-\text{num\_sub}}^{M+2+\text{num\_sub}} \times \alpha_i^{i-\text{num\_sub}+1} \right.$$

$$-\alpha_{i-\text{num\_sub}+1}^{M+2+\text{num\_sub}-1} \times \alpha_i^{i-\text{num\_sub}+2} - \dots \alpha_{i-1}^{M+3} \times \alpha_i^{i}\Big)$$

$$\dots$$

$$\text{num\_sub} = min(i, M-1-(k-1)) \quad \text{number of subtractions for } \alpha_i^{M+k}$$

$$\dots$$

$$\alpha_i^{2M} = \frac{b_i^{2M}}{\mu_i}$$

$$z_i = \frac{y_i - z_0 \times \alpha_i^1 - z_1 \times \alpha_i^2 - \dots - z_{i-1} \times \alpha_i^i}{\mu_i}$$

For $i = M, M+1, \dots, N-1$:

$$\alpha_i^j = b_i^{j-1}, \quad j = 1, 2 \dots, M-1$$

$$\alpha_i^k = \alpha_i^k - \alpha_{i-M}^{M+k-1} \times \alpha_i^1 - \alpha_{i-M+1}^{M+k-2} \times \alpha_i^2 - \dots - \alpha_{i-M+\text{iter}}^{M+k-1-\text{iter}} \times \alpha_i^{1+\text{iter}},$$
$$k = 2, 3, \dots, M, \quad \text{iter} = 0, 1, \dots, k-2, \quad i = M, M+1 \dots, N-1$$

$$\mu_i = b_i^M - \alpha_{i-M}^{2M} \times \alpha_i^1 - \alpha_{i-M+1}^{2M-1} \times \alpha_i^2 - \dots - \alpha_{i-1}^{M+1} \times \alpha_i^M,$$
$$i = M, M+1, \dots, N-1$$

$$\alpha_i^{M+1} = \frac{b_i^{M+1} - \alpha_{i-1}^{M+2} \times \alpha_i^M - \alpha_{i-2}^{M+3} \times \alpha_i^{M-1} - \alpha_{i-3}^{M+4} \times \alpha_i^{M-2} - \dots - \alpha_{i-M+1}^{2M} \times \alpha_i^2}{\mu_i},$$
$$i = M, M+1, \dots, N-2$$

$$\alpha_i^{M+2} = \frac{b_i^{M+2} - \alpha_{i-1}^{M+3} \times \alpha_i^M - \alpha_{i-2}^{M+4} \times \alpha_i^{M-1} - \alpha_{i-3}^{M+5} \times \alpha_i^{M-2} - \dots - \alpha_{i-M+2}^{2M} \times \alpha_i^3}{\mu_i},$$
$$i = M, M+1, \dots, N-3$$

$$\dots$$

$$\alpha_i^{2M} = \frac{b_i^{2M}}{\mu_i}, \qquad i = M, M+1, \dots, N-1-M$$

$$z_i = \frac{y_i - z_{i-M} \times \alpha_i^1 - z_{i-M+1} \times \alpha_i^2 - \dots - z_{i-1} \times \alpha_i^M}{\mu_i}, \qquad i = M, M+1, \dots, N-1$$

and a backward substitution for finding the unknowns x in a reverse order:

$$x_{N-1} = z_{N-1}$$

$$x_{N-k} = z_{N-k} - \alpha_{N-k}^{M+1} \times x_{N-k+1} - \dots - \alpha_{N-k}^{M+1+k-2} \times x_{N-1}, \quad k = 2, 3, \dots, M$$

$$x_i = z_i - \alpha_i^{M+1} \times x_{i+1} - \alpha_i^{M+2} \times x_{i+2} - \alpha_i^{M+3} \times x_{i+3} - \dots - \alpha_i^{2M} \times x_{i+M},$$
$$i = N-(M+1), N-(M+2), \dots, 0.$$

In order to cope with the stability issue of the Thomas method in the case of non-diagonally dominant matrices, in the case of a zero (or numerically zero) quotient of two subsequent leading principal minors within the symbolic method a symbolic variable is assigned instead and the calculations are continued. At the end of the algorithm, this symbolic variable is substituted with zero. The same approach is suggested in [3].

The full multi-diagonal symbolic method in pseudocode is given in Algorithm 1. There, ε plays the role of a numerical zero, and was set to 1.0e−20 in our code.

Algorithm 1: Multi-diagonal symbolic algorithm for solving a SLAE $Ax = y$.

Input: $N, \mathbf{b^0}, \mathbf{b^1}, \dots, \mathbf{b^M}, \mathbf{b^{M+1}}, \dots, \mathbf{b^{2M}}, \mathbf{y}, \varepsilon$
Output: $\mathbf{x}$
if $det(A) == 0$ **then**
  exit
end if
bool flag = False
$\mu_0 = b_0^M$ ▷ Step 1.(0)
if $|\mu_0| < \varepsilon$ **then**
  $\mu_0 =$ symb; flag = True
end if
for $k = \overline{M+1, \dots, 2M}$ **do**
  $\alpha_0^k = \dfrac{b_0^k}{\mu_0}$
end for
$z_0 = \dfrac{y_0}{\mu_0}$
for $k = \overline{1, 2, \dots, M}$ **do**
  $\alpha_1^k = b_1^{k-1}$
end for ▷ (1)
$\mu_1 = b_1^M - \alpha_0^{M+1} \times \alpha_1^M$
if !flag **then**
  if $|\mu_1| < \varepsilon$ **then**
    $\mu_1 =$ symb; flag = True
  end if
end if
for $k = \overline{M+1, \dots, 2M}$ **do**
  $\alpha_1^k = b_1^k$
  if $M > 1$ and $k < 2M$ **then**
    $\alpha_1^k = \alpha_1^k - \alpha_0^{k+1} \times \alpha_1^M$
  end if
  $\alpha_1^k = \dfrac{\alpha_1^k}{\mu_1}$
end for
$z_1 = \dfrac{y_1 - z_0 \times \alpha_1^M}{\mu_1}$
for $i = \overline{2, \dots, N-1}$ **do**
  counter = 0 ▷ number of non-zero helping α_1^k,
  ▷ where $k = \overline{1, 2, \dots, M}$
  if $i < M$ **then**
    counter $= M - i$
  end if
  for $k = \overline{1, \dots, M}$ **do**
    if $k -$ counter ≥ 1 **then**
      $\alpha_i^{k-\text{counter}} = b_i^{k-1}$
    end if
  end for
  ▷ above we shift the non-zero $\alpha_i^j, j \leq M$ in order
  ▷ to have them in the interval $j \in [0; \dots]$
  for $k = \overline{M+1, \dots, 2M}$ **do**
    $\alpha_i^k = b_i^k$
  end for
  $\mu_i = b_i^M$
  $z_i = y_i$
  iter = 0
  ▷ number of iterations for α_i^k, where $k \leq M$
  coeff = 0
  ▷ the biggest distance between the lower coeff of
  ▷ α_i^k and $\alpha_{\text{coeff+iter}}^{M+\dots}$
  if $i >= M$ **then**
    coeff $= i - M$
  end if
  for $k = \overline{2, \dots, M}$ **do**
    iter = 0
    for $l = \overline{2, \dots, k - \text{counter}}$ **do**
      $\alpha_i^{k-\text{counter}} = \alpha_i^{k-\text{counter}}$
        $-\alpha_{\text{coeff+iter}}^{M+k-1-\text{counter}-\text{iter}} \times \alpha_i^{1+\text{iter}}$
      iter = 1 + iter
    end for
  end for
  iter = 0
  ▷ number of iterations for μ_i and z_i
  if $i < M$ **then**
    mu_max_iter $= i - 1$
  else
    mu_max_iter $= M - 1$
  end if
  for iter $= \overline{0, 1, \dots, \text{mu\_max\_iter}}$ **do** ▷ μ_i, z_i
    $\mu_i = \mu_i - \alpha_{\text{coeff+iter}}^{2M-\text{counter}-\text{iter}} \times \alpha_i^{1+\text{iter}}$
    $z_i = z_i - z_{\text{coeff+iter}} \times \alpha_i^{1+\text{iter}}$
  end for
  if !flag **then**
    if $|\mu_i| < \varepsilon$ **then**
      $\mu_i =$ symb; flag = True
    end if
  end if
  $z_i = \dfrac{z_i}{\mu_i}$
  iter = 0 ▷ number of iterations for a
  ▷ particular $\alpha_i^m, m \geq M+1$
  alpha_counter = 0 ▷ number of $\alpha_i^m, m \geq M+1$
  for $m = \overline{0, 1, \dots, M-1}$ **do**
    num_sub$[m] = 0$
    ▷ number of subtractions in α_i^m
  end for
  for $m = \overline{M+1, \dots, 2M-1}$ **do**
    m_index $= m - M - 1$
    ▷ shift the index from 0 to $M - 2$
    if $i \leq M - 1$ **then**

```
            num_sub[m_index] =
                min(i, M − 1 − alpha_counter)
        else
            num_sub[m_index] =
                M − 1 − alpha_counter
        end if
        iter = 0
        for k = 0, 1, ..., num_sub[m_index] − 1 do
            coeff = i − num_sub[m_index] + iter
            coeff_1 = 0
            if i ≥ M then
                coeff_1 = (M − num_sub[m_index]
                                +iter)%M + 1
            else
                coeff_1 = i − num_sub[m_index]
                                +iter + 1
            end if
            ▷ the helping α_i^{coeff_1} are with upper index
            ▷ up to M, therefore we need to find the
                                        ▷ module(M)
            α_i^m = α_i^m
                −α_coeff^{m+num_sub[m_index]−iter} × α_i^{coeff_1}
            iter = iter + 1
        end for
        alpha_counter = alpha_counter + 1
        for k = M + 1, ..., 2M do
            α_i^k = α_i^k / μ_i
        end for
    end for
end for
x_{N−1} = z_{N−1}                                ▷ Step 2. Solution
for i = N − 2, ... 0 do
    x_i = z_i
    iter = 0
    for k = 0, ... M − 1 do
        if i + iter > n − 2 then
            break
        end if
        x_i = x_i − α_i^{M+1+k} × x_{i+1+k}
        iter = iter + 1
    end for
end for
Cancel the common factors in the numerators and
denominators of x, making them coprime. Substi-
tute symb := 0 in 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_i expressions with ε, because any further μ_i is going to be a symbolic expression. To that reason, we use a boolean flag which tells us if any previous μ_i is a symbolic expression. In that case, comparison with ε is not conducted as being not needed.

3. Justification of the algorithm

Let us make some observations on the correctness of the proposed algorithm. In case the algorithm assigns μ_i for any $i = \overline{0, N-1}$ to be equal to a symbolic variable (in case μ_i is zero or numerically zero), this ensures correctness of the formulae for computing the solution of the considered SLAE (because we are not dividing by (numerical) zero). However, this does not add any additional requirements to the coefficient matrix so as to keep the algorithm stable.

Theorem 2. *The only requirement to the coefficient matrix of a multi-diagonal SLAE so as the multi-diagonal symbolic algorithm to be correct 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:

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

because the determinant of an upper triangular matrix is equal to the product of all its diagonal elements [20]. (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:

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

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 $M_{N-1} \neq 0$. □

Remark: above, we have used the following recurrent formula $M_i = \prod_{i=0}^{i} \mu_i$.

Remark: this theorem coincides with the theorem we have proven in [16], because no matter what the number of diagonals $(2 \times M + 1)$ within the coefficient matrix is, the logic remains.

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 only one solution.

3.1. Number of computational steps

The calculation of $\alpha_i^k, \mu_i, \alpha_i^{M+1}, \alpha_i^{M+2}, \dots, \alpha_i^{2M}$, and z_i depends on the results of the calculation of α_{i-j}^{M+k}, and z_{i-j}. On the other hand, the calculation of x_i depends on the results of the calculation of $\alpha_i^{M+1}, \alpha_i^{M+2}, \dots, \alpha_i^{2M}, z_i$, and $x_{i+1}, x_{i+2}, \dots, x_{i+M}$. This makes the multi-diagonal numerical method inherently serial. It takes $2 \times N$ steps overall, where N is the number of equations in the initial SLAE.

3.2. Complexity

The amount of operations per expression are summarized in Table 1. Thus, the overall complexity of the multi-diagonal numerical algorithms is:

$$2NM^2 + 5NM + N - \frac{4M^3}{3} - \frac{7M^2}{2} - \frac{13M}{6},$$

where N is the number of rows in the initial coefficient matrix. Hence, the multi-diagonal numerical method requires only $O(N)$ operations (provided that $M << N$) for finding the solution, and beats the Gaussian elimination which requires $O(N^3)$ operations.

4. Results

Within this paper we formulated the multi-diagonal numerical solver which is a generalization of the Thomas method for multi-diagonal SLAEs. In Algorithm 1, we introduced the pseudocode of the the generalised symbolic algorithm for solving SLAEs with multi-diagonal coefficient matrices.

It was proven that the only requirement to the coefficient matrix of a multi-diagonal SLAE so as the multi-diagonal symbolic algorithm to be correct is nonsingularity.

The multi-diagonal numerical method takes $2 \times N$ steps overall, where N is the number of equations in the initial SLAE.

Table 1

Complexity per expression for the multi-diagonal numerical algorithm

expression	# operations	simplified form of # ops	examples		
			$M=2$	$M=3$	$M=4$
$\alpha_i^k, i<M,$ $k=2,\dots,M$	$\sum_{k=1}^{M-1}(2\times(1+2+\dots k-1)) = \sum_{k=1}^{M-1}\left(2\times\frac{(k-1)\times k}{2}\right)$	$\frac{(M-1)\times M\times(2M-1)}{6} - \frac{(M-1)\times M}{2}$	0	2	8
$\alpha_i^k, i\ge M,$ $k=2,\dots,M$	$(N-1-M+1) \times\sum_{k=1}^{M}((k-1)\times 2)$	$(N-M)\times(M^2-M)$	$2(N-2)$	$6(N-3)$	$12(N-4)$
$\mu_i, i<M$	$\sum_{k=0}^{M-1}(2k)$	M^2-M	2	6	12
$\mu_i, i\ge M$	$(N-1-M+1)\times 2M$	$2MN-2M^2$	$4N-8$	$6N-18$	$8N-32$
$\alpha_i^{M+k}, i<M,$ $k=1,2,\dots,M$	$\sum_{k=1}^{M}\left(\sum_{i=0}^{M-k}(i\times 2+1) +(k-1)\times((M-k)\times 2+1)\right)$	$M^3 - \frac{M\times(M+1)\times(2M+1)}{6} + \frac{M\times(M+1)}{2}$	6	19	44
$\alpha_i^{M+k}, i\ge M,$ $k=1,2,\dots,M$	$\sum_{k=1}^{M}((N-(M+k)) \times((M-k)\times 2+1))$	$NM^2-2M^3-M^2 + \frac{M\times(M+1)\times(2M+1)}{3} - \frac{M\times(M+1)}{2}$	$4N-13$	$9N-41$	$16N-94$
$z_i, i<M$	$\sum_{k=0}^{M-1}(2k+1)$	M^2	4	9	16
$z_i, i\ge M$	$(N-1-M+1)\times(2M+1)$	$2NM+N-M-2M^2$	$5N-10$	$7N-21$	$9N-36$
$x_{N-k},$ $k=1,\dots,M$	$\sum_{k=1}^{M}((k-1)\times 2)$	M^2-M	2	6	12
$x_{N-k},$ $k=M+1,\dots,N$	$(N-(M+1)+1)\times 2M$	$2NM-2M^2$	$4N-8$	$6N-18$	$8N-32$
Total	$2NM^2+5NM+N-\frac{4M^3}{3}-\frac{7M^2}{2}-\frac{13M}{6}$		$19N-29$	$34N-74$	$53N-150$

The complexity of the multi-diagonal numerical algorithms was found to be:

$$2NM^2 + 5NM + N - \frac{4M^3}{3} - \frac{7M^2}{2} - \frac{13M}{6},$$

where N is the number of rows in the initial coefficient matrix. Hence, the multi-diagonal numerical method requires $O(N)$ operations (provided that $M << N$) for finding the solution. The amount of operations per expression were summarized in Table 1. In the Table 1 one can also find the complexity per expression in the cases when $M=2$, $M=3$, and $M=4$.

5. Discussion

Within this paper we formulated the multi-diagonal numerical solver which is a generalization of the Thomas method for multi-diagonal SLAEs. Next, we introduced the pseudocode of the generalised symbolic algorithm for solving SLAEs with multi-diagonal coefficient matrices. There, as a remedy of the stability issue which arises within the Thomas method in the case of non-diagonally dominant matrices, if we obtain a zero (or numerically zero) quotient of two subsequent leading principal minors, a symbolic variable is assigned instead and the calculations are continued. At the end of the algorithm, this symbolic variable is substituted with zero. The generalised multi-diagonal symbolic algorithm is the novelty that addresses the need of a direct method which solves multi-diagonal SLAEs without putting any requirements for the characteristics of the coefficient matrix. This algorithm is a generalization of the algorithms presented in [3, 10, 16].

It was proven that the only requirement to the coefficient matrix of a multi-diagonal SLAE so as the multi-diagonal symbolic algorithm to be correct is nonsingularity. Asking for nonsingularity of the coefficient matrix is a standard requirement so as the SLAE to have only one solution. Hence, this does not limit the significance of the formulated symbolic algorithm.

The multi-diagonal numerical method takes $2 \times N$ steps overall, where N is the number of equations in the initial SLAE.

The multi-diagonal numerical method requires $O(N)$ operations (provided that $M << N$) for finding the solution, and beats the Gaussian elimination which requires $O(N^3)$ operations.

6. Conclusion

A generalised symbolic algorithm for solving systems of linear algebraic equations with multi-diagonal coefficient matrices was formulated and presented in pseudocode. Some notes on the correctness of the algorithm were made. Formula for the complexity of the multi-diagonal numerical algorithm was obtained.

Funding: This research received no external funding.

Data Availability Statement: Data sharing is not applicable.

Acknowledgments: The author would like to thank Dr. Alexander Ayriyan (JINR), and Prof. Toshiyuki Imamura (R-CCS, RIKEN) for their precious comments.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Spiteri, R. J. *Notes in Numerical Analysis I. Chapter 2* (University of Saskatchewan, 2007).
2. Christov, C. I. Gaussian elimination with pivoting for multidiagonal systems. *University of Reading, Department of Mathematics, Numerical Analysis Report 5/94* (1994).
3. El-Mikkawy, M. A Generalized Symbolic Thomas Algorithm. *Applied Mathematics* **3,** 342–345. doi:10.4236/am.2012.34052 (2012).
4. Thomas, L. H. *Elliptic Problems in Linear Difference Equations over a Network* (Watson Sci. Comput. Lab. Rept., Columbia University, 1949).
5. Higham, N. J. *Accuracy and Stability of Numerical Algorithms* 2nd ed. 710 pp. (SIAM, 2002).
6. Samarskii, A. A. & Nikolaev, E. S. *Metody Resheniya Setochnyh Uravnenii* Russian (Nauka, 1978).
7. Samarskii, A. A. & Nikolaev, E. S. *Numerical Methods for Grid Equations. Direct Methods* (Birkhäuser Basel, 1989).
8. Karawia, A. A. & Rizvi, Q. M. On solving a general bordered tridiagonal linear system. *International Journal of Mathematical Sciences* **33** (2013).
9. Atlan, F. & El-Mikkawy, M. A new symbolic algorithm for solving general opposite-bordered tridiagonal linear systems. *American Journal of Computational Mathematics* **5,** 258–266. doi:10.4236/ajcm.2015.53023 (2015).
10. Askar, S. S. & Karawia, A. A. On Solving Pentadiagonal Linear Systems via Transformations. *Mathematical Problems in Engineering. Hindawi Publishing Corporation.* **9.** doi:10.1155/2015/232456 (2015).

11. Jia, J.-T. & Jiang, Y.-L. Symbolic algorithm for solving cyclic penta-diagonal linear systems. *Numerical Algorithms* **63,** 357–367. doi:10.1007/s11075-012-9626-2 (2012).
12. Karawia, A. A. A new algorithm for general cyclic heptadiagonal linear systems using Sherman-Morrisor-Woodbury formula. *ARS Combinatoria* **108,** 431–443 (2013).
13. Bernardin, L. e. a. *Maple Programming Guide* (Maplesoft, a division of Waterloo Maple Inc., 1996-2021.).
14. Wolfram Research, I. *System Modeler* (Version 13.2, Champaign, IL, 2022).
15. Higham, D. J. & Higham, N. J. *MATLAB guide* (SIAM, 2016).
16. Veneva, M. & Ayriyan, A. Symbolic Algorithm for Solving SLAEs with Heptadiagonal Coefficient Matrices. *Mathematical Modelling and Geometry* **6,** 22–29 (2018).
17. Veneva, M. & Ayriyan, A. *Performance Analysis of Effective Methods for Solving Band Matrix SLAEs after Parabolic Nonlinear PDEs* in *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, 2019), 407–419. doi:10.1007/978-3-319-97277-0_33.
18. Veneva, M. & Ayriyan, A. Effective Methods for Solving Band SLAEs after Parabolic Nonlinear PDEs. *AYSS-2017, European Physics Journal – Web of Conferences (EPJ-WoC)* **177** (2018).
19. Veneva, M. & Ayriyan, A. *Performance Analysis of Effective Symbolic Methods for Solving Band Matrix SLAEs* in *23rd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2018)* **214** (European Physics Journal – Web of Conferences (EPJ-WoC), 2019).
20. Kaye, R. & Wilson, R. *Linear Algebra* 242 pp. (Oxford University Press, 1998).

Information about the authors

Milena Veneva—Master of Sciences in Applied Mathematics, (e-mail: milena.p.veneva@gmail.com, ORCID: 0000-0002-6421-4716)

УДК 519.872, 519.217
PACS 07.05.Tp, 02.60.Pn, 02.70.Bf
DOI: 10.22363/2658-4670-2025-33-1-46-56 EDN: BLFUDE

Символьный алгоритм решения СЛАУ с многодиагональными матрицами коэффициентов

Милена Венева[1,2]

[1] Объединённый институт ядерных исследований, ул. Жолио-Кюри, д. 6, Дубна, 141980, Российская Федерация

[2] Центр вычислительной науки RIKEN (R-CCS), Минатодзима-Минами-мачи 7-1-26, район Тюо-ку, Кобе, префектура Хиого, 650-0047, Япония

Аннотация. Системы линейных алгебраических уравнений с многодиагональными матрицами коэффициентов возникают во многих прикладных и теоретических задачах науки и техники, а также в задачах вычислительной линейной алгебры, где их решение представляет собой одну из ключевых проблем. В данной работе представлен обобщённый символьный алгоритм решения систем линейных алгебраических уравнений с многодиагональными матрицами коэффициентов. Алгоритм приведён в виде псевдокода. Сформулирована и доказана теорема, определяющая условие корректности алгоритма. Получена формула, описывающая вычислительную сложность соответствующего численного алгоритма для многодиагональных систем.

Ключевые слова: численные методы, вычислительные методы для разреженных матриц, методы численного математического программирования, вычислительная сложность численных алгоритмов