KDiagonal linear system solver

Pavlov Egor

June 6, 2025

1 Introduction

This text will describe an algorithm for solving systems of linear equations of the form:

$$\begin{bmatrix} a_{00} & a_{01} & ... & a_{0k_2} & 0 & ... & 0 \\ a_{10} & a_{11} & ... & a_{1k_2} & a_{1(k_2+1)} & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ a_{k_10} & a_{k_11} & ... & a_{k_1k_2} & a_{k_1(k_2+1)} & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & 0 & 0 & ... & a_{(N-k_2)(N-1)} \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & 0 & 0 & ... & a_{(N-2)(N-1)} \\ 0 & 0 & ... & 0 & 0 & ... & a_{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ ... \\ ... \\ x_{N-2} \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ ... \\ ... \\ b_{N-2} \\ b_{N-1} \end{bmatrix} \tag{1}$$

For ease of representation and analysis, each diagonal will be considered as a one-dimensional array, which must be supplemented with zero elements until it reaches length N, if we are talking about the upper diagonal, or from the beginning, if we are talking about the lower diagonal:

$$\begin{cases} U_{ji} = a_{i(j+1+i)}, \;\; 0 \le i \le N-1-j, \;\; U_{ji} = 0, \;\; N-j \le i \le N-1, \;\; \text{where } \; 0 \le j \le k_2 - 1 \\ L_{ji} = 0, \;\; 0 \le i \le j, \;\; L_{ji} = a_{(j+1+i)i}, \;\; j+1 \le i \le N-1, \;\; \text{where } \; 0 \le j \le k_1 - 1 \end{cases} \tag{2}$$

Then the system of equations can be rewritten in a simple form:

$$\sum_{l=0}^{k_1-1} L_{li} x_{(i-l)} + a_{ii} x_i + \sum_{l=0}^{k_2-1} U_{li} x_{(i+l)} = b_i, \;\; \text{where } \; 0 \le i \le N-1 \tag{3}$$

In this form, x has indices that go beyond the formal limits of $0,\ N-1$, but this is not so important given the zero values of the coefficients at these points. As mentioned earlier, a_{ii} is not zero, so we divide the corresponding equations by them:

$$\sum_{l=0}^{k_1-1} L_{li} x_{(i-l)} + x_i + \sum_{l=0}^{k_2-1} U_{li} x_{(i+l)} = b_i, \;\; \text{where } \; 0 \le i \le N-1 \tag{4}$$

where used the new definition of U and L:

$$\begin{cases} U_{ji} = a_{i(j+1+i)}/a_{ii}, \;\; 0 \le i \le N-j-1, \;\; U_{ji} = 0, \;\; N-j \le i \le N-1, \;\; \text{where } \; 1 \le j \le k_2 \\ L_{ji} = 0, \;\; 0 \le i \le j, \;\; L_{ji} = a_{(j+1+i)i}/a_{ii}, \;\; j+1 \le i \le N-1, \;\; \text{where } \; 0 \le j \le k_1 \end{cases} \tag{5}$$

We will express x_i in terms of a linear combination of $x_{i+1},\ ..., x_{i+k_2}$:

$$x_i = \sum_{l=0}^{k_2-1} P_{il} x_{i+l+1} + R_i \tag{6}$$

For $i = 0$, the values of P and R are obviously expressed in terms of U and b:

$$P_{0l} = -U_{l0}, \;\; 0 \le l \le k_2 - 1, \;\; R_0 = b_0 \tag{7}$$

For the remaining i, we introduce additional values $Q^i_{(.)(.)}$ and $W^i_{(.)}$ in such a way that:

$$x_{i-l} = \sum_{j=0}^{k_2-1} Q^i_{lj} x_{i+j} + W^i_l \tag{8}$$

Where $0 \leq l \leq k_1$. For $l = 1$:

$$x_{i-1} = \sum_{j=0}^{k_2-1} P_{(i-1)j} x_{i+j} + R_{i-1} \tag{9}$$

Where do we find $Q^i_{1(.)}$ and W^i_1:

$$Q^i_{1j} = P_{(i-1)j}, \ \ 0 \leq j \leq k_2 - 1, \ \ W^i_1 = R_{i-1} \tag{10}$$

Returning to the form (6) and decomposing the sum into two parts (after x_i and before).

$$\begin{aligned} x_{i-l} = \sum_{j=0}^{k_2-1} P_{(i-l)j} x_{i-l+j+1} + R_{i-l} = \\ = R_{i-l} + \sum_{j=l}^{k_2-1} P_{(i-l)j} x_{i-l+j+1} + P_{(i-l)(l-1)} x_i + \sum_{j=0}^{l-2} P_{(i-l)j} x_{i-l+j+1} \end{aligned} \tag{11}$$

Now we use (8) for x in the second sum:

$$\begin{aligned} x_{i-l} = R_{i-l} + \sum_{j=0}^{k_2-1-l} P_{(i-l)(l+j)} x_{i+j+1} + P_{(i-l)(l-1)} x_i + \\ + \sum_{j=0}^{l-2} P_{(i-l)j} \Big(\sum_{p=0}^{k_2-1} Q^i_{(l-j-1)p} x_{i+p} + W^i_{l-j-1} \Big) \end{aligned} \tag{12}$$

As a result, we get the expression for $Q^i_{l(.)}$ and W^i_l through $Q^i_{(l-p)(.)}$, W^i_{l-p}, $P_{(i-p)(.)}$ and R_{i-p} where $1 \leq p$:

Now let's go back to the form (4), rewriting it as:

$$x_i = b_i - \sum_{l=0}^{k_2-1} U_{li} x_{(i+l+1)} - \sum_{l=0}^{k_1-1} L_{li} x_{(i-l-1)} = b_i - \sum_{l=0}^{k_2-1} U_{li} x_{(i+l+1)} - \sum_{l=0}^{k_1-1} L_{li} \Big(\sum_{j=0}^{k_2-1} Q^i_{(l+1)j} x_{i+j} + W^i_{(l+1)} \Big) \tag{13}$$

Moving all x_i to the left side, we find the new $P_{(i)(.)}$ and R_i:

$$x_i \Big(1 + \sum_{l=0}^{k_1-1} L_{li} Q^i_{(l+1)0} \Big) = b_i - \sum_{l=0}^{k_2-1} U_{li} x_{(i+l+1)} - \sum_{l=0}^{k_1-1} L_{li} \Big(\sum_{j=1}^{k_2-1} Q^i_{(l+1)j} x_{i+j} + W^i_{(l+1)} \Big) \tag{14}$$

For $i = N - 1$:

$$x_{N-1} = \Big(b_{N-1} - \sum_{l=0}^{k_1-1} L_{l(N-1)} W^{N-1}_{(l+1)} \Big) / \Big(1 + \sum_{l=0}^{k_1-1} L_{l(N-1)} Q^{N-1}_{(l+1)0} \Big) \tag{15}$$

Knowing x_{N-1} and $P_{i(.)}$, R_i you can restore all x_i in reverse order. Thus, to calculate N values, it is necessary to calculate the order of $\sim k_1 k_2$ auxiliary values at each step, hence the total complexity of the algorithm:

$$O(N k_1 k_2) \tag{16}$$

Since the usual algorithms for direct solution of systems of linear equations have complexity $O(N^3)$, the resulting algorithm is efficient for any k.