Review of Cyclic Reduction for Parallel Solution of Hermitian Positive Definite Block-Tridiagonal Linear Systems

Martin Neuenhofen

July 3, 2018

Abstract

Cyclic reduction is a method for the solution of (block-)tridiagonal linear systems. In this note we review the method tailored to hermitian positive definite banded linear systems.

The reviewed method has the following advantages: It is numerically stable without pivoting. It is suitable for parallel computations. In the presented form, it uses fewer computations by exploiting symmetry. Like Cholesky, the reviewed method breaks down when the matrix is not positive definite, offering a robust way for determining positive definiteness.

Brief Summary

Equations (9) give formulas to separate the block-tridiagonal linear system (1) into two separated block-tridiagonal systems of half the dimension. This can be used for solving the system in a parallel divide-and-conquer approach. The resulting method is described in Algorithm 1.

1 Hermitian Block-Tridiagonal Linear Systems

We skip the introductory section and literature review on cyclic reduction methods and move directly to the problem.

We consider the linear system

$$
\underbrace{\begin{bmatrix}
\mathbf{A}_1 & \mathbf{B}_1^{\mathsf{H}} & & & & \\
\mathbf{B}_1 & \mathbf{A}_2 & \mathbf{B}_2^{\mathsf{H}} & & & \\
 & \mathbf{B}_2 & \mathbf{A}_3 & \mathbf{B}_3^{\mathsf{H}} & & \\
 & & \mathbf{B}_3 & \ddots & \ddots & \\
 & & & \ddots & \ddots & \mathbf{B}_{N-1}^{\mathsf{H}} \\
 & & & & \mathbf{B}_{N-1} & \mathbf{A}_N
\end{bmatrix}}_{=:\underline{\mathbf{A}}}
\cdot
\underbrace{\begin{pmatrix}
\mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \vdots \\ \mathbf{x}_N
\end{pmatrix}}_{=:\underline{\mathbf{x}}}
=
\underbrace{\begin{pmatrix}
\mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \mathbf{y}_4 \\ \vdots \\ \mathbf{y}_N
\end{pmatrix}}_{=:\underline{\mathbf{y}}},
\tag{1}
$$

where the matrices

$$\mathbf{A}_j \in \mathbb{C}^{m \times m} \qquad\qquad \forall\, j = 1, ..., N \qquad\qquad (2a)$$

$$\mathbf{B}_j \in \mathbb{C}^{m \times m} \qquad\qquad \forall\, j = 1, ..., (N-1) \qquad\qquad (2b)$$

$$\mathbf{y}_j \in \mathbb{C}^{m \times k} \qquad\qquad \forall\, j = 1, ..., N \qquad\qquad (2c)$$

are given. $m, k \in \mathbb{N}$ are integers that shall be much smaller than $N \in \mathbb{N}$. The matrix

$$\underline{\mathbf{A}} \in \mathbb{C}^{(N \cdot m) \times (N \cdot m)}$$

shall be positive definite. Sought are the matrices

$$\mathbf{x}_j \in \mathbb{C}^{m \times k} \qquad\qquad \forall\, j = 1, ..., N\,. \qquad\qquad (3)$$

We assume $N \in 2 \cdot \mathbb{N}$.

2 Cyclic Reduction Method for Hermitian Positive Definite Systems

Derivation of the Method   The idea of cyclic reduction is based on a special reordering. Putting the elements of $\underline{\mathbf{A}}$ in the order

$$1, 3, 5, 7, \ldots, (N-1)\,,\; 2, 4, 6, 8, \ldots, N\,, \qquad\qquad (4)$$

we obtain the rearranged matrix

$$
\left[
\begin{array}{ccccc|ccccc}
\mathbf{A}_1 & & & & & \mathbf{B}_1^{\mathsf{H}} & & & & \\
& \mathbf{A}_3 & & & & \mathbf{B}_2 & \mathbf{B}_3^{\mathsf{H}} & & & \\
& & \mathbf{A}_5 & & & & \mathbf{B}_4 & \mathbf{B}_5^{\mathsf{H}} & & \\
& & & \ddots & & & & \ddots & \ddots & \\
& & & & \mathbf{A}_{N-1} & & & & \mathbf{B}_{N-2} & \mathbf{B}_{N-1}^{\mathsf{H}} \\
\hline
\mathbf{B}_1 & \mathbf{B}_2^{\mathsf{H}} & & & & \mathbf{A}_2 & & & & \\
& \mathbf{B}_3 & \mathbf{B}_4^{\mathsf{H}} & & & & \mathbf{A}_4 & & & \\
& & \mathbf{B}_5 & \ddots & & & & \mathbf{A}_6 & & \\
& & & \ddots & \mathbf{B}_{N-2}^{\mathsf{H}} & & & & \ddots & \\
& & & & \mathbf{B}_{N-1} & & & & & \mathbf{A}_N
\end{array}
\right]\,.
$$

The four sub-blocks in the above matrix we denote with $\underline{\mathbf{D}}_1, \underline{\mathbf{D}}_2, \underline{\mathbf{C}} \in \mathbb{C}^{(N/2 \cdot m) \times (N/2 \cdot m)}$. I.e., for the above matrix we use the compact writing

$$
\left[
\begin{array}{c|c}
\underline{\mathbf{D}}_1 & \underline{\mathbf{C}}^{\mathsf{H}} \\
\hline
\underline{\mathbf{C}} & \underline{\mathbf{D}}_2
\end{array}
\right]\,.
$$

These three matrices $\underline{\mathbf{D}}_1, \underline{\mathbf{D}}_2, \underline{\mathbf{C}}$ can be used to construct two separate linear systems for the solution components $\mathbf{x}_j$. As a benefit, the two separated linear

systems have only half the dimension of the original system. In particular, one of the two systems is only in the odd indices $j = 1, 3, 5, \ldots, (N-1)$, while the other one is only in the even indices for $\mathbf{x}_j$. We clarify this in the following.

To be able to write the systems in compact form, we use the notation of even and odd vectors

$$
\underline{\mathbf{x}}_o = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_3 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_{N-1} \end{pmatrix}, \quad
\underline{\mathbf{x}}_e = \begin{pmatrix} \mathbf{x}_2 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{pmatrix}, \quad
\underline{\mathbf{y}}_o = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_3 \\ \mathbf{y}_5 \\ \vdots \\ \mathbf{y}_{N-1} \end{pmatrix}, \quad
\underline{\mathbf{y}}_e = \begin{pmatrix} \mathbf{y}_2 \\ \mathbf{y}_4 \\ \mathbf{y}_5 \\ \vdots \\ \mathbf{y}_N \end{pmatrix}.
$$

The decoupled linear systems for $\underline{\mathbf{x}}_0$ and $\underline{\mathbf{x}}_e$ are:

- Odd System:

$$
\underbrace{(\underline{\mathbf{D}}_1 - \underline{\mathbf{C}}^\mathsf{H} \cdot \underline{\mathbf{D}}_2^{-1} \cdot \underline{\mathbf{C}})}_{=:\underline{\mathbf{U}}} \cdot \underline{\mathbf{x}}_o = \underbrace{\underline{\mathbf{y}}_o - \underline{\mathbf{C}}^\mathsf{H} \cdot \underline{\mathbf{D}}_2^{-1} \cdot \underline{\mathbf{y}}_e}_{=:\underline{\mathbf{u}}}
\tag{5}
$$

- Even System:

$$
\underbrace{(\underline{\mathbf{D}}_2 - \underline{\mathbf{C}} \cdot \underline{\mathbf{D}}_1^{-1} \cdot \underline{\mathbf{C}}^\mathsf{H})}_{=:\underline{\mathbf{V}}} \cdot \underline{\mathbf{x}}_e = \underbrace{\underline{\mathbf{y}}_e - \underline{\mathbf{C}} \cdot \underline{\mathbf{D}}_1^{-1} \cdot \underline{\mathbf{y}}_o}_{=:\underline{\mathbf{v}}}
\tag{6}
$$

The odd system has the system matrix $\underline{\mathbf{U}}$ and right-hand side $\underline{\mathbf{u}}$. The even system has $\underline{\mathbf{V}}$ and $\underline{\mathbf{v}}$. In the following we give formulas for these matrices and vectors.

Formulas for the Odd and Even System We now give explicit formulas for the matrices $\underline{\mathbf{U}}, \underline{\mathbf{V}} \in \mathbb{C}^{(N/2 \cdot m) \times (N/2 \cdot m)}$ and the right-hand sides $\underline{\mathbf{u}}, \underline{\mathbf{v}} \in \mathbb{C}^{(N/2 \cdot m) \times k}$, that build the odd and even system.

The matrices and vectors have the block-structure below:

$$
\underline{\mathbf{U}} = \begin{bmatrix}
\mathbf{U}_1 & \mathbf{E}_1^\mathsf{H} & & & & \\
\mathbf{E}_1 & \mathbf{U}_2 & \mathbf{E}_2^\mathsf{H} & & & \\
& \mathbf{E}_2 & \mathbf{U}_3 & \mathbf{E}_3^\mathsf{H} & & \\
& & \mathbf{E}_3 & \ddots & \ddots & \\
& & & \ddots & \ddots & \mathbf{E}_{N/2-1}^\mathsf{H} \\
& & & & \mathbf{E}_{N/2-1} & \mathbf{U}_{N/2}
\end{bmatrix}, \quad
\underline{\mathbf{u}} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \vdots \\ \mathbf{u}_{N/2-1} \\ \mathbf{u}_{N/2} \end{pmatrix} ; \tag{7a}
$$

$$
\underline{\mathbf{V}} = \begin{bmatrix}
\mathbf{V}_1 & \mathbf{F}_1^\mathsf{H} & & & & \\
\mathbf{F}_1 & \mathbf{V}_2 & \mathbf{F}_2^\mathsf{H} & & & \\
& \mathbf{F}_2 & \mathbf{V}_3 & \mathbf{F}_3^\mathsf{H} & & \\
& & \mathbf{F}_3 & \ddots & \ddots & \\
& & & \ddots & \ddots & \mathbf{F}_{N/2-1}^\mathsf{H} \\
& & & & \mathbf{F}_{N/2-1} & \mathbf{V}_{N/2}
\end{bmatrix}, \quad
\underline{\mathbf{v}} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \vdots \\ \mathbf{v}_{N/2-1} \\ \mathbf{v}_{N/2} \end{pmatrix} . \tag{7b}
$$

There appear the inner matrices

$$\mathbf{U}_j, \mathbf{V}_j \in \mathbb{C}^{m \times m} \qquad \forall\, j = 1, 2, \ldots, (N/2) \qquad (8a)$$

$$\mathbf{E}_j, \mathbf{F}_j \in \mathbb{C}^{m \times m} \qquad \forall\, j = 1, 2, \ldots, (N/2 - 1) \qquad (8b)$$

$$\mathbf{u}_j, \mathbf{v}_j \in \mathbb{C}^{m \times k} \qquad \forall\, j = 1, 2, \ldots, (N/2). \qquad (8c)$$

The formulas for the these inner matrices can be found by insertion. We present these formulas below:

$$\mathbf{U}_j = \mathbf{A}_{2 \cdot j - 1} - \mathbf{B}_{2 \cdot j - 2} \cdot \mathbf{A}_{2 \cdot j - 2}^{-1} \cdot \mathbf{B}_{2 \cdot j - 2}^{\mathsf{H}} - \mathbf{B}_{2 \cdot j - 1}^{\mathsf{H}} \cdot \mathbf{A}_{2 \cdot j}^{-1} \cdot \mathbf{B}_{2 \cdot j - 1} \qquad (9a)$$

$$\mathbf{E}_j = -\mathbf{B}_{2 \cdot j} \cdot \mathbf{A}_{2 \cdot j}^{-1} \cdot \mathbf{B}_{2 \cdot j - 1} \qquad (9b)$$

$$\mathbf{u}_j = \mathbf{y}_{2 \cdot j - 1} - \mathbf{B}_{2 \cdot j - 2} \cdot \mathbf{A}_{2 \cdot j - 2}^{-1} \cdot \mathbf{y}_{2 \cdot j - 2} - \mathbf{B}_{2 \cdot j - 1}^{\mathsf{H}} \cdot \mathbf{A}_{2 \cdot j}^{-1} \cdot \mathbf{y}_{2 \cdot j} \qquad (9c)$$

$$\mathbf{V}_j = \mathbf{A}_{2 \cdot j} - \mathbf{B}_{2 \cdot j - 1} \cdot \mathbf{A}_{2 \cdot j - 1}^{-1} \cdot \mathbf{B}_{2 \cdot j - 1}^{\mathsf{H}} - \mathbf{B}_{2 \cdot j}^{\mathsf{H}} \cdot \mathbf{A}_{2 \cdot j + 1}^{-1} \cdot \mathbf{B}_{2 \cdot j} \qquad (9d)$$

$$\mathbf{F}_j = -\mathbf{B}_{2 \cdot j + 1} \cdot \mathbf{A}_{2 \cdot j + 1}^{-1} \cdot \mathbf{B}_{2 \cdot j} \qquad (9e)$$

$$\mathbf{v}_j = \mathbf{y}_{2 \cdot j} - \mathbf{B}_{2 \cdot j - 1} \cdot \mathbf{A}_{2 \cdot j - 1}^{-1} \cdot \mathbf{y}_{2 \cdot j - 1} - \mathbf{B}_{2 \cdot j}^{\mathsf{H}} \cdot \mathbf{A}_{2 \cdot j + 1}^{-1} \cdot \mathbf{y}_{2 \cdot j + 1} \qquad (9f)$$

The matrices on the left can be computed in parallel for $j = 1, 2, 3, \ldots, (N/2)$.

It is important that in the above we have – for the sake of compact formulas – used identical equations for all j. However, for $j = 1$ for instance, the formula for $\mathbf{U}_1$ accesses the matrix $\mathbf{B}_0$, that does not exist. To repair this detail, we define the following auxiliary matrices, that appear in the above formulas:

$$\mathbf{B}_0 := \mathbf{0} \in \mathbb{C}^{m \times m}, \qquad\qquad \mathbf{B}_N := \mathbf{0} \in \mathbb{C}^{m \times m}$$

$$\mathbf{A}_0 := \mathbf{I} \in \mathbb{C}^{m \times m}, \qquad\qquad \mathbf{A}_{N+1} := \mathbf{I} \in \mathbb{C}^{m \times m}$$

Formulation of the Method With the above ingredients, we are able to describe how cyclic reduction is used to solve the linear system (1). The method consists of three phases:

In the first phase, given the data (2), the method uses the formulas in (9) to compute the block matrices (8) for the odd system (5) and even system (6).

In the second phase, cyclic reduction is used recursively for solving the odd and even system for their respective solutions $\underline{\mathbf{x}}_o, \underline{\mathbf{x}}_e$. I.e., we consider (5) and (6) as instances of (1). This works excellent for the following reason: We assumed that $\underline{\mathbf{A}}$ is positive definite. If the assumption holds, then further it holds that $\underline{\mathbf{U}}, \underline{\mathbf{V}}$ are positive definite as well, because they are Schur complements of $\underline{\mathbf{A}}$.

In the third phase, we rearrange the vectors $\underline{\mathbf{x}}_o, \underline{\mathbf{x}}_e$ into the global solution vector $\underline{\mathbf{x}}$ of (1). This completes the method.

Determination of Positive Definiteness If the matrix $\underline{\mathbf{A}}$ is indeed positive definite, then all recursive calls of cyclic reduction will succeed, until at the bottom there arise positive definite linear systems of size $m \times m$. These shall be solved with Cholesky's method.

If, in contrast to the assumption, the matrix $\underline{\mathbf{A}}$ is not positive definite, then eventually at least either of the matrices $\underline{\mathbf{U}}, \underline{\mathbf{V}}$ will not be well-defined, and thus the method will crash. This crash will appear in the form that Cholesky's method determines indefiniteness for one of the matrices $\mathbf{A}_j$ in (9).

3 Description of a Parallel Method

Computing Model   For our description of the parallel method, we consider a parallel computing system that consists of N computing nodes; each has an index, counted as $1, 2, 3, ..., N$. The nodes are connected in a communication network, that consists of cables. The length each cable equals the distance of the respective nodes in space. Information travels along these cables at a limited speed.

Idea of the Parallel Method   The idea for a parallel cyclic reduction consists of two ingredients:

1. **Parallel instantiation of odd and even systems:** The equations (9) are evaluated in parallel over $j = 1, \ldots, (N/2)$. If N parallel computing units are given, then the nodes $1, 2, \ldots, N/2$ can compute the matrices for the odd system, and the nodes with indices $N/2 + 1, \ldots, N$ can compute the matrices for the even system.

2. **Parallel recursive solution of the split systems:** The linear systems (5) and (6) are solved recursively for $\underline{\mathbf{x}}_o$ and $\underline{\mathbf{x}}_e$. To this end, the same method as for (1) can be used.

A pseudo-code for this procedure is given in Algorithm 1. The algorithm assumes $N \in 2^{\mathbb{N}}$.

Time-Complexity and Communication   The pure time-complexity of parallel computations is in

$$\mathcal{O}(\log(N) \cdot m^3).$$

But, there is an additional cost for communication of data between the computing nodes. This cost is discussed below.

The method has significant non-local communication. We describe this in the following, where j ranges from 1 to $N/2$:

For the computation of the odd system, the node j receives data from nodes $2 \cdot j - 2$, $2 \cdot j - 1$, and $2 \cdot j$. Analogously, for the computation of the even system, the node $N/2 + j$ receives data from nodes $2 \cdot j - 1$, $2 \cdot j$, and $2 \cdot j + 1$.

Depending on the communication network of the parallel computing system, this non-local communication can be very expensive. In fact, it may result in a time complexity that exceeds the time spent in actual parallel computations.

Non-local communications are not unavoidable in parallel methods for solving block-tridiagonal linear systems. In [1] we provide an accurate description of

Algorithm 1 Cyclic Reduction Method

1: **procedure** Solver($\underline{\mathbf{A}}, \underline{\mathbf{y}}$)
2:     **if** $\dim(\underline{\mathbf{A}})$ is sufficiently small **then**
3:         Solve $\underline{\mathbf{A}} \cdot \underline{\mathbf{x}} = \underline{\mathbf{y}}$ for $\underline{\mathbf{x}}$, using Cholesky's method.
4:         **return** $\underline{\mathbf{x}}$
5:     **end if**
6:     *// parallel for-loop for nodes $j = 1, 2, \ldots, (N/2)$*
7:     **for** $j = 1, 2, \ldots, N/2$ **do**
8:         Compute $\mathbf{U}_j, \mathbf{E}_j, \mathbf{u}_j$ from formulas (9).
9:     **end for**
10:    *// concurrent parallel for-loop for nodes $j = (N/2)+1, (N/2)+2, \ldots, N$*
11:    **for** $j = 1, 2, \ldots, N/2$ **do**
12:        Compute $\mathbf{V}_j, \mathbf{F}_j, \mathbf{v}_j$ from formulas (9).
13:    **end for**
14:    *// The even and odd systems $\underline{\mathbf{U}}, \underline{\mathbf{V}}$ are given as $\mathbf{U}_j, \mathbf{E}_j, \mathbf{V}_j, \mathbf{F}_j, \mathbf{u}_j, \mathbf{v}_j$ according to (7)*
15:    $\underline{\mathbf{x}}_o := $Solver($\underline{\mathbf{U}}, \underline{\mathbf{u}}$)   *// solve on nodes $j = 1, 2, \ldots, (N/2)$*
16:    $\underline{\mathbf{x}}_e := $Solver($\underline{\mathbf{V}}, \underline{\mathbf{v}}$)   *// solve on nodes $j = (N/2)+1, (N/2)+2, \ldots, N$*
17:    Reassemble $\underline{\mathbf{x}}$ from $\underline{\mathbf{x}}_o, \underline{\mathbf{x}}_e$.
18:    **return** $\underline{\mathbf{x}}$
19: **end procedure**

---

a method (with directly implementable code for an MPI cluster) that works only via local communications. The communication network used for this method is a binary tree – the cheapest possible way for connecting all nodes.

4 Numerical stability

If $\underline{\mathbf{A}}$ is positive definite, then: The matrices $\underline{\mathbf{D}}_1, \underline{\mathbf{D}}_2$ are regular, and the matrices $\underline{\mathbf{Q}}_o$ and $\underline{\mathbf{Q}}_e$ are positive definite. If $\underline{\mathbf{A}}$ is positive definite, and Cholesky decomposition with forward and backward substitution is used to compute formulas (9), then: The matrices $\underline{\mathbf{Q}}_o$ and $\underline{\mathbf{Q}}_e$ are bit-wise hermitian, and the algorithm is numerically stable. In particular, the substitution must be applied as follows:

Given a formula like

$$\mathbf{Z} := \mathbf{G}^{\mathsf{H}} \cdot \mathbf{M}^{-1} \cdot \mathbf{G}$$

we must use the Cholesky decomposition $M = \mathbf{L} \cdot \mathbf{L}^{\mathsf{H}}$ in the following way:

1: $\mathbf{L} := $Chol($\mathbf{M}$)
2: $\tilde{\mathbf{G}} := \mathbf{L} \backslash \mathbf{G}$
3: **return** $\mathbf{Z} := \tilde{\mathbf{G}}^{\mathsf{H}} \cdot \tilde{\mathbf{G}}^{\mathsf{H}}$

Using the cyclic reduction in a recursive way, or solving the split systems with an arbitrary forward stable numerical algorithm, results in an overall method that is numerically stable.

The described algorithm can also be used to determine whether a matrix **A** at hand is positive definite. Namely, this is the case if and only if all split subsystems are positive definite. Hence, we can stop the recursive splitting when the submatrices are small enough, and then check, e.g. with Cholesky, for the positive definiteness of **U**, **V**.

References

[1] M. P. Neuenhofen. A time-optimal algorithm for solving (block-)tridiagonal linear systems of dimension n on a distributed computer of n nodes. *ArXiv e-prints. DOI:1801.09840*, jan 2018.