|  |  |
| --- | --- |
|  |  |

The Ministry of Education and Science of the Russian Federation

Lobachevsky State University of Nizhni Novgorod

Computing Mathematics and Cybernetics faculty

The competitiveness enhancement program   
of the Lobachevsky State University of Nizhni Novgorod   
among the world's research and education centers

Strategic initiative  
“Achieving leading positions in the field of supercomputer technology   
and high-performance computing”

**DIRECT METHODS   
FOR SOLVING SYSTEM OF LINEAR EQUATIONS**

*Lecture 4. Sweep and reduction methods for solving systems   
with tridiagonal matrix*

Nizhni Novgorod

2014

# Objectives

The objective of the lecture is the investigation of the direct methods for solving linear equations systems with band matrix (by the example of the tridiagonal matrix) and also the research of the approaches to their efficient parallel implementation in shared memory systems.

# Abstract

The sweep and reduction methods for solving systems of *n* linear algebraic equations with a tridiagonal matrix *A* are considered. The classical variants of the algorithms are stated. The estimates for the algorithms error that arises when calculating with machine precision are shown. Two variants of the sweep method (right and left sweep) are investigated. Combination of these two variants makes it possible to implement efficient parallelization for two threads. For parallelization in general case the block sweep variant is stated. A variant of parallel algorithm for the reduction method is also demonstrated. It may be applied in general case as well.

# GUIDELINES

The methods for solving systems of *n* linear algebraic equations with a tridiagonal matrix *A* and a right-hand side *b* are investigated. The problem of solving a linear equations system with the given matrix *A* and the vector *b* is considered to be a problem of searching a value of the unknown vector *x* such that all the equations of the system hold.

One of the important categories of the linear equations systems is the systems with band matrix. Let us consider the methods for solving such problems by the example of the systems with tridiagonal matrix.

One of the well-known methods for solving the system of this type is the sweep method. It consists of two steps: a forward phase (calculation of the auxiliary coefficients *αi*, *βi*) and a backward phase (calculation of the values of the unknowns *xi* with the help of *αi*, *βi*). It is known that the fulfillment of the *diagonal dominance* condition is necessary for the computational stability of the sweep method. The sweep method complexity amounts to 8*n*+*O*(1).

If the values of the unknowns *xi* are determined from right to left, the method is said to be the *right sweep*. Analogously, the formulas of the *left sweep* may be written: the values of the unknowns *xi* are determined from left to right.

Combination of the left and right sweeps gives us the *counter-sweep* method which makes the parallelization for two threads possible. For this, the system is distributed among the threads: the right sweep is performed in the first thread, the left one is performed in the second thread. In the equation with number *n*/2 the solving processes are interfaced, the value of *xn*/2 is determined and, after that, each thread determines its own part of the unknown vector independently.

Research now a parallelization scheme of the sweep method when *p* threads are used. Apply the block approach to the data partitioning: let each thread process *m*=*n*/*p* rows of the matrix *A*. Within a matrix stripe that is handled by the *k*-th thread the elimination of the subdiagonal elements of the matrix could be organized (the forward phase of the method). If the elimination of the subdiagonal variables by the first thread does not add any new coefficients to the matrix, the elimination of the subdiagonal variables in the other threads will lead to generation of the column of the nonzero coefficients: in all the blocks (except the first one) a number of the nonzero elements in row will not change but the equations structure will change. After that, the backward phase of the algorithm is performed: each thread eliminates the superdiagonal elements starting with the last one. After the backward phase completion, the matrix becomes a block one. Eliminate the inner rows of every stripe. As a result, an equations system relative to a part of initial unknowns is derived. Such system contains 2*p* equations and is tridiagonal. It could be solved by the sequential sweep method. After this system is solved, the values of the unknowns on the boundaries of the data division stripes will become known. Then, the values of the inner variables could be determined during one step.

The described method of parallelization already gives us the good results. However, a better strategy for the elimination of the unknowns may be used. The forward phase of a new algorithm is the same while, during the backward phase, every thread eliminates superdiagonal elements starting with its last but one element and finishing in the last one for the previous thread. The order change for the unknowns elimination in the backward phase of the algorithm gives us an opportunity to generate an auxiliary task of the smaller size. Let us eliminate all the rows within every stripe of the matrix except the last one. As a result, an equations system relative to a part of initial unknowns is derived. Such system contains just *p* equations and is also tridiagonal. It could be solved by the sequential sweep method. After this system is solved, the values of the unknowns on the lower boundaries of the data division stripes will become known. Then, the values of the inner variables could be determined in every thread during one step.

The theoretical speedup estimate that could be achieved under such approach amounts to *p*/2.

The analysis of the formulas for the sweep method shows that in some cases method could result in great error. A potential error source is the formulas for calculating the “sweep” coefficients, which include a division operation by the difference between close values.

The *reduction method* that will be discussed below does not have such disadvantage and, moreover, when being realized on the modern computational systems, it demonstrates the greater efficiency in comparison with the sweep method. A main restriction of the reduction method consists in the fact that it is applied only for the matrices of the size equal to a power of two. A generalization of the reduction method to the case of block tridiagonal matrices exists. In such case a number of the blocks must be equal to a power of two and the main idea remains the same (only the division operation is substituted by the multiplication operation by an inverse matrix; this substitution supposes a good invertibility of the blocks of the matrix *A*).

The idea of the reduction method consists in two following steps. The *forward phase* of the method is a sequential elimination of the unknowns from the source system: first, the unknowns with the odd numbers, then – with the numbers multiple to 2 (but not multiple to 4), etc. The *backward phase* of the method is a restoration of the values of the odd variables based on the known values of the even variables. The complexity of the reduction method amounts 11*n*+*O*(1) operations.

The analysis of the computational scheme of the reduction method shows that every next step of the forward and backward phase depends on the previous one. The forward and backward phases could not be wholly parallelized successfully. On the other hand, the elimination of the unknowns on the specific step of the forward phase could be performed independently since there is no data dependency in such case. The specific step of the backward phase could be parallelized in the similar way since the values of the unknowns are determined independently.

The computational experiments for estimating the speedup of the parallel sweep and reduction methods were carried out both when solving an only equation with a big matrix and solving an equations system with a matrix of small size and various right-hand sides.

# RECOMMENDATIONS for students

The questions of solving the band equations systems (including the solving of the tridiagonal systems) are discussed in [1] in detail. The cyclic reduction method (with both sequential and parallel variants) is investigated in [2]. The program implementation of the sequential algorithms is stated in [3].

# References

1. Gene H. Golub, Charles F. Van Loan. Matrix Computations. The John Hopkins University Press, 1996.
2. Golub, G. H., Ortega, J. M.: Scientific Computing. An Introduction with Parallel Computing. Academic Press, Inc., 1993.
3. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. Numerical Recipes. The Art of Scientific Computing. Cambridge University Press, 2007.

# EXERCISES

1. Implement the parallel counter-sweep method for solving the linear equations systems with tridiagonal matrix when using two threads. Estimate the speedup in comparison with the sequential program version.
2. Implement the block sweep method for solving the linear equations systems with tridiagonal matrix. Estimate the speedup in comparison with the sequential program version.
3. Implement the parallel cyclic reduction method for solving the linear equations systems with tridiagonal matrix. Compare the method performance with the performance of the sweep method. Estimate the speedup in comparison with the sequential program version.

# TEST QUESTIONS

1. What is the complexity of the sweep method?
   1. + 8*n* + *O*(1)
   2. 8*n*2 + *O*(*n*)
   3. 8*log(*n*)* + *O*(1)
2. Specify the condition of the computational stability for the sweep method
   1. The matrix must be symmetric
   2. +The matrix must be diagonally dominant
   3. The matrix must be nonsingular
3. The counter-sweep method
   1. is poorly parallelizable, there is no speedup.
   2. + is simply parallelizable for the systems of any size.
   3. is simply parallelizable but only in case of systems of small size.
4. The counter-sweep method
   1. + could be efficiently parallelized only for two threads;
   2. could be efficiently parallelized for *p*>2 threads;
   3. could not be efficiently parallelized.
5. Choose a preferable format for storing a tridiagonal matrix of size *n*×*n*.
   1. A two-dimensional array of size *n*×*n*
   2. + Three arrays of length *n*, which correspond to three diagonals
   3. *n* arrays of length 3, which correspond to nonzero elements in the columns of the matrix
6. An equations system with a tridiagonal matrix of size *n*×*n* is solved using the parallel counter-sweep method. What number of the operations is performed by each thread?
   1. 4*n*+ *O*(1)
   2. *n*/4+ *O*(1)
   3. *n*2*/4*+ *O*(*n*)
7. What is the complexity of the reduction method?
   1. + 11*n*+*O*(1)
   2. 11*n*2+*O*(*n*)
   3. 11*log(*n*)* + *O*(1)
8. The reduction method could be applied for the systems with tridiagonal matrix
   1. of any size
   2. of size multiple to 2
   3. + of size multiple to 2*n*
9. In the reduction method
   1. + the parallelization is possible only within one step of the forward or backward phase;
   2. the forward or backward phase could be parallelized entirely;
   3. the parallelization is not possible, the method is not parallelizable.