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

Computational mathematics and cybernetics faculty

APPROVED

Dean of the Computational mathematics
and cybernetics faculty

_________________ V. Gergel

"_____"___________________2014.

Course program

DIRECT METHODS FOR SOLVING SYSTEM OF LINEAR EQUATIONS

Bachelor program "Fundamental Informatics and Information Technologies"

Specialization

General part

Professional block

Elective disciplines

Б3.В.ДВ.9

Degree

Bachelor

Education form

Full-time attendance

Nizhni Novgorod

2014

1. Course objective

The course examines the known numerical algorithms for solving systems of linear algebraic equations, and also a range of issues related to parallelization of these algorithms in shared memory systems.

The course is based on the materials developed in UNN with the support of Intel in 2011-2013 years. (http://www.hpcc.unn.ru/?doc=491). In 2014, a partial modification for the course materials was made as a part of the UNN competitiveness enhancement program. The changes mainly consist in the detailed planning of individual student work. In addition, the key course elements were translated into English.

The main course objective is to study the base algorithms for solving systems of linear algebraic equations, form skills of developing parallel numerical algorithms focused on efficient application in shared memory systems.

It involves solving the following **problems**:

1. Mastering the parallel programming technologies for shared memory systems (OpenMP, TBB, Cilk Plus).
2. Investigation of the basic matrix operations as well as problems of their efficient implementation for memory hierarchy systems.
3. Studying the algorithms for solving systems of linear equations with both matrices of general type and matrices of special form (Gaussian method, Cholesky factorization, sweep and reduction methods)
4. Consideration of the general approaches to algorithm optimization for memory work and load balancing for parallelization.
5. Discussing (within the laboratory works) the examples of efficient implementation of the examined algorithms.

The course targets engineers, teachers and research assistants as well as postgraduate students and students in higher education.

2. Course position in the bachelor program

The course is developed for the 4-th year students and is given in the 8-th semester. This course belongs to the elective disciplines of the professional block.

The lecture part of the course is oriented on the students with basic skills of the linear algebra and numerical methods in accordance with the courses from the Bachelor’s program.

The students are supposed to have basic skills of designing C/C++ programs and parallel programming using OpenMP in order to perform the laboratory works. Competence in TBB and MKL as well as the other components of the Intel Parallel Studio XE package is encouraged, but not mandatory for studying the course. All the information about using Intel Parallel Studio XE is included in the course materials.

3. Learning outcomes and requirements

In the framework of this course, the following **competencies** are formed:

- Possessing the general culture of thinking, the ability to perceive, compile and analyze information (General Competency 1 - GC1). Students will be able to:
    - construct oral and written arguments in a logical and clear manner.
- The ability of intellectual, cultural, moral, physical, and professional self-development and self-improvement (GC 2). Students will be able to:
    - constantly improve their professional and cultural level.
- The ability to understand and apply in practice the theory of information as a fundamental scientific basis of information technology (Professional Competency 1 – PC 1). Students will be able to:
    - understand the content side of the information process, know the techniques for sending, receiving, processing, analyzing and storing data.
- The ability to understand, develop and apply modern information technology (PC 4). Students will be able to:
    - understand the concepts and implement the functionality of the following core technologies:
        - at the level of technological literacy:
            - computer systems architecture;
        - at the level of in-depth knowledge:
            - basic programming;
            - parallel and distributed computing.
    - develop and use professionally the software for supporting information systems and processes, to be able to use modern instrumental computing equipment.
- The ability to conduct research (PC 5). Students will be able to:
    - develop new algorithmic and methodological and technological solutions;

    - collect, process and interpret the data of modern research necessary to develop approaches, decisions and conclusions on appropriate scientific and professional issues.
- The ability to conduct analytical activities (PC 8). Students will be able to:
    - analyze and select modern technologies and methodologies for implementing an information system.

As a result of education graduates from the course will know and be able to:

- apply the parallel algorithms for solving systems of linear equations with both matrices of general type and matrices of special form;
- analyze and decompose into parts the algorithms allowing the parallel execution;
- develop the parallel programs for shared memory computing systems using the technologies OpenMP, TBB, Cilk Plus;
- conduct computational experiments on the high performance computing systems;
- estimate efficiency of the performed parallel computations.

4. Course outline

The course consists of 1 credit, 36 hours, including 12 lecture hours and 8 practice hours. Practice classes can be held as lab works (students carry out assignments step-by-step under supervision) or master class (supervisor demonstrates and explains step-by-step solutions). 16 hours are allocated for individual work. The authors encourage additional work.

4.1. Course outline

Course outline is as follows:

#	**Module**	**Semester**	**Week**	**Module type**				**Assessment**
				Lecture	Seminar	Lab	Individual work	
1	Implementation of the standard matrix algorithms in shared memory systems	6	1	2	–	–	1	Test
2	Gaussian elimination method for solving systems of general type.		2	2	–	–	1	Test
3	Cholesky factorization for solving systems with		3-4	2	–	0	2	Test

	symmetric positive definite matrix							
4	Sweep and reduction methods for solving systems with tridiagonal matrix		4-5	2	–	–	1	Test
5	Application of sweep and reduction methods for solving linear equation system with band matrix by example of compound option pricing problem		6-7	–	–	2	2	Test
6	Sparse matrix storage formats, implementation of standard matrix operations		7-8	2	–	–	2	Test
7	Sparse matrix-vector multiplication. Loop parallelization with OpenMP, TBB, ArBB, Cilk Plus.		9-10	–	–	2	2	Test
8	Algorithmic optimization in sparse algebra problems by example of matrix multiplication		11-12	–	–	2	2	Test
9	Solving of systems of linear algebraic equations with sparse matrix by example of Cholesky factorization		13-14	2	–	–	1	Test
10	Solving of sparse SLAE by direct methods in heat conduction problem in a plate. MKL PARDISO usage.		15-17	–	–	2	2	Test
	TOTAL:			**12**	–	**8**	**16**	**Final assessment form – exam**

4.2. Course description

Course content is as follows:

1. **Implementation of the standard matrix algorithms in shared memory systems**

   The objective of this lecture is to investigate the standard matrix algorithms (by example of matrix-vector and matrix-matrix multiplication) and also study of the approaches to their efficient parallel implementation in shared memory systems

2. **Gaussian elimination method for solving systems of general type**

   Investigation of the direct methods for solving linear equations systems with square matrix and also the research of the approaches to their efficient parallel implementation in shared memory systems

3. **Cholesky factorization for solving systems with symmetric positive definite matrix**

Investigation of the direct methods for solving linear equations systems with symmetric positive definite matrix and also the research of the approaches to their efficient parallel implementation in shared memory systems

4. **Sweep and reduction methods for solving systems with tridiagonal matrix**
   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
5. **Application of sweep and reduction methods for solving linear equation system with band matrix by example of compound option pricing problem**
   In the given laboratory work a linear algebraic equation system with a tridiagonal matrix is investigated. For solving such system the special methods are known such as, for example, the *sweep* and *cyclic reduction methods* considered in this work. The cyclic reduction method is a little more complex to implement, but rounding errors influence on this method less than on the sweep method. The problems of the cyclic reduction method parallelization in the shared memory systems are discussed in this work.
6. **Sparse matrix storage formats, implementation of standard matrix operations**
   Investigation of the sparse matrix storage formats and also the algorithms implementing standard matrix operations. The efficiency of the proposed algorithms that take into account matrix sparseness is emphasized.
7. **Sparse matrix-vector multiplication. Loop parallelization with OpenMP, TBB, ArBB, Cilk Plus.**
   The objective of this work is to investigate some parallel programming technologies in shared memory systems by example of sparse matrix-vector multiplication problem.
8. **Algorithmic optimization in sparse algebra problems by example of matrix multiplication**
   The objective of this work is to investigate basic principles of sparse matrices storage and some processing algorithms for them.
9. **Solving of systems of linear algebraic equations with sparse matrix by example of Cholesky factorization**
   Investigation of the direct methods for solving system of linear equations with sparse matrix by example of Cholesky factorization
10. **Solving of sparse SLAE by direct methods in heat conduction problem in a plate. MKL PARDISO usage.**

The objective of the given laboratory work is to investigate some methods for solving system of linear equations with a sparse matrix by example of stationary problem of heat distribution in a rectangular plate with specified temperature boundary conditions.

5. Learning technologies

During course we use the following learning technologies: lectures, lab works, individual work, assessment techniques. Powerpoint presentations for all lectures and practical lessons are used.

6. Individual work and assessment techniques

Individual work consists of mastering theoretical and practical material according to the given references, solving practical problems, and answering on the given questions. Individual work can be done in both classes and at home. Control of individual work is performed by electronic tests. In the end of the program there is a final test.

6.1 Assessment forms

Monitoring of progress in studies is performed by tests in class that consist of assignments from the list of questions and practical problems (given below).

The final attestation is done based on the results of the final test. This test includes questions from all sections of the course.

6.2 Individual work: Questions and Practical problems

Module 1

1. Implement the parallel matrix-vector multiplication algorithm based on block matrix decomposition. Derive theoretical estimates of the time consumed by the algorithm taking into account the parameters of the used computing system. Carry out the computing experiments. Compare the results of real experiments with the theoretical estimates derived before.
2. Design the implementation of block algorithms for matrices multiplication that might be executed for rectangular grids of the threads of general form.

Module 2

1. Implement the block algorithm for solving the linear equations system with a triangular matrix and several right-hand sides $LX=B$.
2. Implement LU-decomposition for a square matrix.
3. Implement the parallel block LU-decomposition for a square matrix.

Module 3

1. Implement the block algorithm for solving the linear equations system with a triangular matrix and several right-hand sides $LX=B$.
2. Implement the Cholesky factorization for a symmetric positive definite matrix.
3. Implement the parallel block Cholesky factorization for a symmetric positive definite matrix.

Module 4

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.

Module 5

1. In order to solve a tridiagonal system, use the built-in function of the MKL library. Estimate the efficiency of using of the library functions in comparison with the sweep method and the sequential implementation for the cyclic reduction method.
2. Implement the counter-sweep method. Estimate the efficiency of using of the counter-sweep method in comparison with the usual sweep and the parallel implementation of the cyclic reduction method. Explain the derived results of the experiments.
3. Implement the block sweep method. Estimate the efficiency of using it in comparison with all the previous implementations of the methods for solving tridiagonal systems. Explain the derived results of the experiments. Estimate the scalability of the performed implementation of the block sweep.

Module 6

1. Develop a sequential implementation for the algorithm of multiplication of sparse matrix by dense vector.
2. Based on the sequential program develop a parallel implementation using the OpenMP technology.
3. Estimate a cache misses number in the OpenMP-implementation when increasing threads number, in order to clarify the reasons for the non-linear scalability. Use Intel Parallel Studio XE tool.

Module 7

1. Develop parallel implementation of the dense matrices multiplication algorithm using the technologies OpenMP, Cilk and TBB.

2. Design two implementations for the scalar multiplication of the dense vectors using the functions call and map of the ArBB library.
3. Develop the dense matrices multiplication algorithm using the ArBB library. Remark: consider the matrices multiplication problem as the repeated calculation of the scalar products of the first matrix rows and the second matrix columns.
4. Estimate the number of cache misses when increasing the number of the threads in the OpenMP-implementation in order to find out the reasons for the nonlinear scalability. Use the instrument Intel Parallel Studio XE.
5. Explain the reason for the absence of the linear speedup of the parallel implementations developed with OpenMP, Cilk and TBB.
6. Design a generator for the matrices of any other structure (e.g., with the increasing number of the nonzero elements) and carry out the experiments with all developed implementations.

Module 8

1. Carry out the experiments on multiplication of the matrices in the CCS format. Reveal and explain the effects concerning correlation of the execution times for the different sequential algorithms. Compare with the base version presented in the work (matrices in the CRS format). Develop and set up a parallel implementation.
2. Carry out the experiments on multiplication of the matrices in the coordinate format. Reveal and explain the effects concerning correlation of the execution times for the different sequential algorithms. Compare with the base version presented in the work (matrices in the CRS format). Develop and set up a parallel implementation.
3. Carry out the experiments on multiplication of the matrices of another structure. Consider matrices with the same number of elements per row. Reveal and explain the effects concerning correlation of the execution times for the different sequential algorithms. Develop and set up a parallel implementation.
4. Adapt the algorithms used in the work to rectangular matrices. Develop a software implementation. Carry out the computational experiments.

Module 9

1. Develop an implementation of constructing a permutation matrix by means of the minimum degree method. Carry out the computational experiments on the matrices with the different patterns.

2. Develop an implementation of constructing a permutation matrix by means of the nested dissection method. Carry out the computational experiments on the matrices with the different patterns.
3. Implement the Cholesky factorization for the sparse matrices (supposing that the matrix is already reordered by means of one of the previously implemented methods). Carry out the computational experiments on the matrices with the different patterns. Compare the fill-in coefficient of factor without reordering and the one with reordering.

Module 10

1. Estimate the complexity of the Cholesky method for the case of five-diagonal matrix considered in the work.

1. Estimate the complexity of the phases of SLAE solving (reordering, symbolic and numerical factorization, etc.)

2. Compute analytically the fill-in of the factor for the heat conduction matrix without reordering and compare it with the fill-in value calculated by PARDISO when factoring the matrix with a direct permutation.

3. Implement the Gaussian method for solving SLAE with a five-diagonal matrix considered in the work (both sequential and parallel versions). Perform the scalability analysis for the application using the parallel implementation of the Gaussian method.

6.4 Assessment criteria

Perfect	Correct answers on >95% of the number of test questions
Excellent	Correct answers on 80-95% of the number of test questions
Very good	Correct answers on 70-79% of the number of test questions
Good	Correct answers on 60-69% of the number of test questions
Satisfactory	Correct answers on 50-59% of the number of test questions
Unsatisfactory	Correct answers on 25-49% of the number of test questions
Bad	Correct answers on <25% of the number of test questions

7. References

1. Alan George, Joseph W. H. Liu. Computer solution of large sparse positive definite systems. Prentice-Hall, 1981.

2. Sergio Pissanetzky. Sparse matrix technology. Academic Press, 1984.

3. David R. Kincaid and E. Ward Cheney. Numerical analysis : mathematics of scientific computing. Brooks/Cole Publishing Company, 1991.

4. Golub, G. H., Ortega, J. M.: Scientific Computing. An Introduction with Parallel Computing. Academic Press, Inc., 1993.
5. Gene H. Golub, Charles F. Van Loan. Matrix Computations. The John Hopkins University Press, 1996.
6. James W. Demmel. Applied Numerical Linear Algebra. SIAM, 1997.
7. Richard L. Burden, J. Douglas Faires. Numerical Analysis. Brooks Cole, 2000.
8. Joe D. Hoffman. Numerical Methods for Engineers and Scientists. CRC Press, 2001.
9. Butcher J.C. Numerical Methods for Ordinary Differential Equations. John Wiley & Sons, 2003.
10. G. Karniadakis, R. Kirby. Parallel scientific computing in C++ and MPI. Cambridge university press, 2003.
11. M. Quinn. Parallel programming in C with MPI and OpenMP. McGraw-Hill, 2004.
12. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. Numerical Recipes. The Art of Scientific Computing. Cambridge University Press, 2007.
13. Intel Math Kernel Library https://software.intel.com/en-us/intel-mkl/

8 Course support

The following software and hardware are used during course study:

Hardware

"Lobachevsky" supercomputer with the Intel Xeon Phi coprocessors is used.

Software

Intel Parallel Studio XE (C/C++ Compiler for Intel Xeon Phi, Intel Amplifier, Intel MKL) is used.

9 Authors

English version of the course is developed by K. Barkalov. V. Gergel, E. Kozinov, A. Sysoyev, A. Pirova, V. Kustikova participated in the Russian version preparation.

Associate professor ____ Konstantin Barkalov

Course program is discussed by Software Department members.

«______» ________________ 2014; Document # _______

The head of the Software Department, prof. ____________________________ Roman Strongin

Course program is approved by methodical commission of Computational Mathematics and Cybernetics Faculty of UNN,

«______» _________________ 2014; Document # _______

The head of the commission _________________________________ Natalia Shestakova