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”

ITERATIVE METHODS FOR SOLVING LINEAR SYSTEMS

Practice 6. Solving sparse linear systems using the preconditioned biconjugate gradient method

Nizhni Novgorod

2014

OBJECTIVES

The purpose of this laboratory work is to demonstrate practical implementation of the biconjugate gradient method and study influence of preconditioning on the method convergence rate.

ABSTRACT

In the course of this laboratory work, we shall review one of iterative methods of solving linear systems, i.e. the biconjugate gradient (BiCG) method. In the course of work, it will be proposed to implement this method for linear systems with non-symmetric sparse matrices. The condition number μ_A of the linear system matrix will have a decisive influence on the method convergence rate, so our next task will consist in studying the method convergence depending on μ_A. Experiments prove poor convergence for ill-conditioned problems. Then it will be proposed to use the $ILU(p)$-preconditioner to improve conditioning of the linear system matrix and analyze the preconditioned method convergence.

GUIDELINES

In the course of this laboratory work, it is proposed to study the biconjugate gradient (BiCG) method. Just like the GMRes method, the BiCG method is intended for linear systems with a $n \times n$ nonsingular matrix A. At the same time, unlike GMRes, BiCG requires less memory but computes products expressed as $A^T x$ which ensures efficient access to the matrix A columns.

The first part of this work will be theoretical. It is known that the matrix $A^T A$ is a symmetric positive definite and the system $A^T Ax=A^T b$ is equivalent to the initial one. Theoretically, the new system can be solved using the conjugate gradient system. However, this will involve matrix multiplication so the matrix $A^T A$ may no longer be sparse. Plus, the matrix $A^T A$ is much worse conditioned than the matrix A. If μ_A is the matrix A condition number, the matrix $A^T A$ condition number will be equal to $(\mu_A)^2$.

The conjugate gradient method cannot be used immediately; the biconjugate gradient method implements another idea. The orthogonal residual sequence is replaced by two biorthogonal sequences $\{r_1,\ldots,r_m\}$ and $\{\bar{r}_1,\ldots,\bar{r}_m\}$. The sequence of conjugate directions is replaced by the sequence of biconjugate directions $\{p_1,\ldots,p_m\}$ and $\{\bar{p}_1,\ldots,\bar{p}_m\}$. This enables easy method formulation based on the conjugate gradient method and give the method computation formulas.

During the following part of work, students will perform a sequential implementation of the BiCG method. For this purpose, the solution **BiCGMethod** is created to contain the sequential biconjugate gradient method implementation and eventuallly that of the preconditioned biconjugate gradient method.

For convenience, the biconjugate gradient method implementation is divided into three projects:

- **parser** is the project containing implementation of functions that enables reading of system from file and a number of operations required for memory allocation and data initialization.
- **routine** is the project containing some mathematic operations such as multiplication of matrices and vectors and checking the solution for consistency.
- **BiCG** is the project containing the biconjugate gradient method implementation and the program main function.

For sparse matrix representation, use the CRS (Column Row Storage) format. This format involves matrix storage in three arrays: **`Value`** is the array containing nonzeroes recorded row-by-row, **`Col`** is the array for the nonzero column indices, **`RowIndex`** is the array for matrix row indices in the **`Col`** array.

The main program function will perform the following sequence of actions:

- Reading the command line arguments
- Reading the system matrix from the file
- Initialization of variables
- Memory allocation
- Setting the right-hand vector
- Solving linear systems using the biconjugate gradient method
- Computation of the system residual over the obtained solution.
- Method operation data output
- Memory release

Experimental results for the biconjugate gradients method demonstrate a significant influence of the machine arithmetics on the convergence rate. It is known than in the precise arithmetics, the biconjugate gradient method must converge in no more than n iterations, where n is the linear system matrix dimension. However, practical implementation shows that some (ill-conditioned) problems require more than n iterations for the method to converge.

As for other methods, preconditioning is required to improve the convergence rate. A preconditioner is a matrix M that has the following properties:

- M is a nonsingular matrix
- $M^{-1}A$ is a well-conditioned matrix

- Matrices M and M^T are easily inversible (which means that the systems $Mz=r$ and $M^Tz=r$ are easy to solve)

The $ILU(p)$ preconditioner which is an incomplete LU-factorization will be used for the purpose of this work (see the laboratory work on preconditioning methods).

As part of this laboratory work, computational formulas for the preconditioned BiCG method will be given and students will be proposed to implement this method.

First of all, add the project containing the $ILU(p)$ preconditioner implementation to the solution. After that, one can implement **`BiCG_M()`**, a new fucntion of solving linear systems using the preconditioned biconjugate gradient method. First of all, copy all parameters from the BiCG() function prototype to the prototype. Then, add to the function prototype structures of the multiplier matrices L and U as inputs and implement the function in accordance with the algorithm computational formulas.

Experimental results for the preconditioned biconjugate gradients method demonstrate a significant improvement of the method convergence rate.

RECOMMENDATIONS FOR STUDENTS

A brief method description including algorithm pseudocode can be found in [1]. A detailed description of the method including the respective conclusions, theoretical justification and application examples can be found in [2].

REFERENCES

1. J. Dongarra et al. Templates for the solution of linear systems: building blocks for iterative methods. SIAM, 1994.
2. Y. Saad. Iterative Methods for Sparse Linear Systems. SIAM, 2003.

PRACTICE

1. Analyze the BiCG method convergence rate depending on precision of the floating point arithmetic. Use the float, double and long double data types and the mpfr library of real numbers.
2. Implement the BiCG-Stab method or stabilized biconjugate gradient method using the pseudocode indicated in [1]. Compare convergence rates of the initial and stabilized methods.

TEST

1. What systems are solvable using the biconjugate gradient method?
    a. Only systems with a symmetric positive definite matrix
    b. Only systems with a symmetric matrix
    c. + Systems with a general matrix
2. What is the main difference of the biconjugate gradient method compared to the conjugate gradient method?
    a. +This method is applicable to non-symmetric matrices.
    b. This method has a greater accuracy.
    c. This method has a greater convergence rate.
3. What is the principal disadvantage of the biconjugate gradient method?
    a. +The matrix AA^T has a greater condition number than A
    b. The matrix AA^T has a smaller condition number than A
4. What is the complexity of a single biconjugate gradient method iteration?
    a. $O(n)$
    b. +$O(n^2)$
    c. $O(n^3)$
5. What are the main operations of the biconjugate gradient method?
    a. +Scalar multiplication
    b. + Matrix multiplication by a vector
    c. Matrix transposing
6. What are possible reasons for a greater biconjugate gradient method error?
    a. +Poor matrix condition
    b. +Numeric computational error
    c. Solving linear systems with a non-symmetric matrix
7. Preconditioning of the biconjugate gradient method
    a. +Improves the numerical method accuracy
    b. Reduces the numerical method accuracy
8. Preconditioning of the biconjugate gradient method
    a. Improves the theoretical method accuracy
    b. Reduces the theoretical method accuracy
    c. +Has no influence on the theoretical method accuracy
9. Preconditioning of the biconjugate gradient method enables
    a. +Reduction of the method iterations count to reach the required accuracy
    b. +Imrpovement of the method computational stability

10. Can the method acceleration become close to the linear acceleration if the main mathematic operations of the method are parallelized?
    a. Yes
    b. +No