



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”

## **NUMERICAL METHODS FOR SOLVING DIFFERENTIAL EQUATIONS**

*Lecture 3. Solving Partial Differential Equations as Illustrated by Wave  
Equation and Heat Equation*

Nizhni Novgorod

2014

## OBJECTIVES

The purpose of this lecture is to study methods of numerical solution of partial differential equations and construction of corresponding parallel algorithms. This is illustrated by the wave equation and heat transfer problem.

## ABSTRACT

The lecture is dedicated to solving second-order partial differential equations using the finite difference method. Solving the wave and heat equations is given in detailed description. For each example, the lecture states the problem, determines the difference scheme, describes the way of creating computation architecture and lists the experimental results.

## GUIDELINES

The lecture is dedicated to solving second-order partial differential equations and parallel numerical methods of solving them. This class of problems is important as mathematical models of many processes in mechanics (vibration of strings, rods, diaphragms and 3D shapes), physics (electromagnetic oscillations, heat transfer or particle diffusion in a medium), fluid and gas dynamics (various types of flows) are described by the second-order equations.

The introductory part of this lecture defines main notions of this field such as partial differential equations and their order, linear and quasilinear differential equations, differential equation solution, initial and boundary conditions and their types. In practice, only a limited class of problems can be solved analytically. Most problems describing phenomena and processes around us allow for a numerical solution only.

A universal method of solving the above differential equations is the *finite difference method* of which confines solving differential equations to solving difference equations. To convert a differential equation to a difference one, replace the continuous domain with a discrete set of points (*grid*) and replace the differential equation with a difference one. The following problems typical for methods of finite differences in general arise in connection with construction of a difference scheme. First, one has to make sure that the linear system has only one solution and indicate the algorithm that makes it possible to obtain it. Second, one has to show that if the grid step tends to zero, the difference problem solution will converge to the initial differential problem solution.

Let us study the first Dirichlet boundary value problem for a one-dimensional wave equation

$$\frac{\partial^2 u}{\partial t^2} = \alpha^2 \frac{\partial^2 u}{\partial x^2} + f(x, t) \text{ in the domain } 0 < x < l, 0 < t \leq T \text{ with the boundary conditions } u(x, 0) = u_0(x),$$

$$0 \leq x \leq l, u(0, t) = \mu_1(t), u(l, t) = \mu_2(t), 0 \leq t \leq T. \text{ To solve it, it is proposed to introduce a}$$

uniform grid with the step  $h$  in the direction  $x$  and step  $\tau$  in the direction  $t$  followed by construction of a difference scheme based on a five-point stencil. The resulting difference scheme will use three layers in time and have second-order error for  $\tau$  and  $h$  at the interior points.

An explicit difference scheme will solve the defined problem. To ensure computational stability of this scheme, fulfillment of the CFL condition  $\alpha\tau < h$  is necessary and sufficient. The solution is obtained in a layer-wise manner: the grid function values on the new layer can be computed after obtaining its values on the previous layer. Thus, for the purposes of parallel algorithm construction, computation of values on the next layer may be used as a subproblem allowing for parallelizing. A set of space points from one layer is distributed among threads. Each thread computes values for its set of points. At the same time, as opposed to distributed memory systems, shared memory systems are not prone to the points passing problem at the bounds of data distribution. To perform computational experiments, the test problem

$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2}$ ,  $x \in [0,1], t \in [0,2]$  was formulated with the boundary conditions  $u(0,t) = u(1,t) = 0$ ,

$u(x,0) = 0.1 \sin(\pi x)$ ,  $\frac{\partial u}{\partial x}(x,0) = 0$ . In order to satisfy the computational stability condition, we

parted the time into  $m=25000$  parts and varied number of the space parts  $n$  from 2,500 to 10,000. The experimental results showed that the effected parallel implementation gave the best acceleration for 4 or 5 threads (3.5 to 4.1 times); however, further increase in the number of threads reduced the speed up. This can be explained by low complexity of operations performed by each thread. In this case, overhead (time required to create and close a parallel section) outweighs the benefits of parallelization. To confirm this, an experiment was performed in the context of increasing computational load: the system equation was replaced by

$\frac{\partial^2 u}{\partial t^2} = \frac{\exp(\cos(x))}{10} \frac{\partial^2 u}{\partial x^2}$ . For the purposes of difference scheme construction, the partition

number  $n$  for the variable  $t$  was set the same as in the first experiment. The experiments showed that the speed up of a parallel version in the context of increased computational load was subject to ramping if the number of threads ranged from 1 through 7; the maximum speed up ranged from 6.2 through 6.9 times.

Now let us discuss the process of heat transfer in a one-dimension homogeneous rod  $0 < x < l$ , described by the equation  $\frac{\partial u}{\partial t} = \alpha^2 \frac{\partial^2 u}{\partial x^2} + f(x,t)$ . Let us study the first Dirichlet

boundary value problem in the domain  $0 < x < 1$ ,  $0 < t \leq T$  with the boundary conditions  $u(x,0) = u_0(x)$ ,  $0 \leq x \leq 1$ ,  $u(0,t) = u_1(t)$ ,  $u(1,t) = u_2(t)$ ,  $0 \leq t \leq T$  and the equation coefficient  $\alpha = 1$ . To solve it, it is proposed to introduce a uniform grid with the step  $h$  in the direction  $x$  and step  $\tau$  in the direction  $t$  followed by construction of a finite difference scheme based on a six-point stencil. This lecture features the following difference scheme for the points within the region:

$$\frac{v_i^{j+1} - v_i^j}{\tau} = \sigma \frac{v_{i+1}^{j+1} - 2v_i^{j+1} + v_{i-1}^{j+1}}{h^2} + (1 - \sigma) \frac{v_{i+1}^j - 2v_i^j + v_{i-1}^j}{h^2} + \varphi_i^j$$
 (here,  $v_i^j$  is a grid function which is

the exact solution of the difference scheme that approximates the exact solution of the differential equation within grid points;  $\sigma$  is the weight parameter and  $\varphi_i^j$  is some right-hand part). Let us consider some special cases of the scheme depending on the parameter.

If  $\sigma = 0$ , this scheme will be explicit. The scheme will be computationally stable if  $\tau \leq h^2/2$  (conditional stability). In this case, the scheme will approximate the initial equation with the order  $O(\tau+h^2)$ . The disadvantage of this scheme is that it requires serious computational effort as the computational stability condition sets specific requirements to the time step.

If  $\sigma = 1$ , the scheme becomes implicit. The system of equations constructed using the computational scheme is tridiagonal and may be solved using the tridiagonal matrix algorithm. The implicit difference scheme is computationally stable for any relation between the steps  $\tau$  and  $h$  (absolute stability). The approximation error is  $O(\tau+h^2)$ .

Specific attention is given to the case where  $\sigma = 1/2$ : in this case, this scheme is also called the *Crank–Nicolson scheme*. Such scheme is absolutely stable with the approximation error of  $O(\tau^2+h^2)$ . Thanks to these properties, the scheme with the weight  $1/2$  will be preferable for computation as in this case the accuracy will be high enough while the grid step will not be too small.

The algorithm of heat transfer problem parallelization is the same as for the wave equation. The subproblems solved in parallel will consist in computation of values on the following layers. If a fully implicit scheme or the Crank–Nicolson scheme is used, computation of the grid function value on the next layer consists in solving a linear system with a tridiagonal matrix. To solve this system, a parallel version of the tridiagonal matrix algorithm is used.

In this lecture, a differential equation in the following form is solved

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 C}{\partial z^2} + \left(r - D - \frac{1}{2}\sigma^2\right) \frac{\partial C}{\partial z} - rC = 0$$

Such equations are used in mathematical finance to determine prices of convertible options (here,  $C(z, t)$  is the convertible option price at the time  $t$ ). See a detailed problem description and the difference scheme for its numerical solution in the practice “Solving Partial Differential Equations”.

The test problem was solved using the Crank–Nicolson scheme, the partition number in the  $t$  variable was 100 while the partition number in the  $z$  variable ranged from 256 through 32,768. The resulting tridiagonal system was solved using the tridiagonal matrix algorithm and the parallel tridiagonal matrix algorithm. Experimental results showed that the use of the parallel tridiagonal matrix algorithm ensures speed up to 1.85 times for 8 threads during problem solution. To solve the problem, no allowance was made for the specific character of tridiagonal systems as they have equal numbers on the diagonals. If this peculiarity is taken into account, it will result in a greater speed up.

## RECOMMENDATIONS FOR STUDENTS

See [Ошибка! Источник ссылки не найден., Ошибка! Источник ссылки не найден., Ошибка! Источник ссылки не найден.] for a detailed description of numerical methods of solving differential equations. See [1] for the methods of solving linear systems and the tridiagonal matrix algorithm in particular.

## REFERENCES

1. Golub G.H., Van Loan Ch. F. Matrix Computations. The John Hopkins University Press, 1996.
2. Hoffman J.D. Numerical Methods for Engineers and Scientists, 2nd Edition. New York: CRC Press, 2001.
3. Kincaid D.R., Cheney E.W. Numerical Analysis: Mathematics of Scientific Computing, 3rd Edition. Pacific Grove: Brooks Cole, 2001.

## PRACTICE

1. Implement the algorithm of solving the wave equation with Type 1 boundary conditions. Parallelize computations by layer and calculate the speed up.
2. Use the algorithm to solve the heat equation with Type 1 boundary conditions using the difference scheme with the weight  $\sigma = 0$  and  $\sigma = 1$ . Compare the computational error for the two weight values. Parallelize computations by layer and calculate the speed up.

## TEST

1. What second-order differential equation problem specifies normal derivative values on the boundary?
  - a. Dirichlet problem
  - b. + Neumann problem
  - c. Robin problem

2. What is the approximation order for the initial condition  $\frac{\partial u}{\partial t}(x,0) = \bar{u}_0(x)$  in case of the wave equation numerical solution using the difference scheme described in the lecture? Step  $h$  in the  $x$  variable and step  $\tau$  in the  $t$  variable.
  - a. First in the  $\tau$  variable and second in the  $h$  variable
  - b. Second in the  $\tau$  variable and first in the  $h$  variable
  - c. + Second in the  $\tau$  and  $h$  variable
3. What is the necessary and sufficient stability condition of the explicit difference scheme for numerical solution of the wave equation discussed above?
  - a.  $\alpha h > \tau$
  - b. +  $\alpha \tau < h$
  - c.  $\alpha \tau > h$
4. How can one parallelize the wave equation solution using the described explicit difference scheme?
  - a. + Perform parallel computations in the points of the same grid layer
  - b. Perform parallel computations in the points of adjacent grid layers
  - c. Perform wave-scheme based computations starting from lower partition indices in the  $t$  and  $x$  variables
5. What is the reason for low scalability of the proposed parallel version of solving the wave equation?
  - a. Synchronizations between the threads
  - b. Data race
  - c. + Low computation load of threads
6. What is the necessary and sufficient condition of computational stability of the explicit difference scheme for solution of a heat equation whose weight is 0?
  - a. +  $\tau \leq h^2/2$
  - b.  $\tau < h^2/2$
  - c.  $\tau \leq h/2$
7. What is the approximation error of the implicit difference scheme for solution of a heat equation whose weight is 1?
  - a.  $O(\tau^2 + h^2)$
  - b. +  $O(\tau + h^2)$
  - c.  $O(\tau^2 + h)$
8. What is the advantage of the implicit difference scheme whose weight is 1 compared to the explicit difference scheme whose weight is 0 for numerical solution of the heat equation?
  - a. It has a higher solution approximation order for the interior grid points
  - b. + Its computational stability does not depend on the partition number in the  $t$  and  $x$  variables
  - c. It has a higher solution approximation order for the interior grid points; its computational stability does not depend on the partition number in the  $t$  and  $x$  variables
9. What is the approximation error of the implicit difference scheme for solution of a heat equation whose weight is 1/2?
  - a. +  $O(\tau^2 + h^2)$ .

b.  $O(\tau+h^2)$ .

c.  $O(\tau^2+h)$ .

10. How can one parallelize the heat equation solution if the Crank-Nicolson scheme is used?

a. Use the grid computation wavefront starting from higher partition indices in the  $t$  and  $x$  variables

b. Use the grid computation wavefront starting from lower partition indices in the  $t$  and  $x$  variables

c. + Use the parallel tridiagonal matrix algorithm