International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:6, No:4, 2012

Parallel Algorithm for Numerical Solution of Three-Dimensional Poisson Equation

Alibek Issakhov

***Abstract*—**In this paper developed and realized absolutely new algorithm for solving three-dimensional Poisson equation. This equation used in research of turbulent mixing, computational fluid dynamics, atmospheric front, and ocean flows and so on. Moreover in the view of rising productivity of difficult calculation there was applied the most up-to-date and the most effective parallel programming technology - MPI in combination with OpenMP direction, that allows to realize problems with very large data content. Resulted products can be used in solving of important applications and fundamental problems in mathematics and physics.

***Keywords*—**MPI, OpenMP, three dimensional Poisson equation

I. Introduction

PARTIAL differential equations are widely used for modeling in various fields of mathematical science, engineering and fluid dynamics. Unfortunately, the explicit solution of these equations analytically is possible only in special simple cases, and as a result, the ability to analyze mathematical models, based on differential equations, is provided by the approximate numerical solution methods. The volume of calculations performed at the same time is cumbersome.

There are problems which cannot be mass produced in personal computers within a reasonable computational time. The solution of these problems is using multiprocessor computer systems with vast array architectures. For parallel computing in multiprocessor computer systems is necessary to create special programs. The text of this program is determined by the parts that can be executed in parallel, and by the algorithm of their interaction.

Problems to solve the full three-dimensional incompressible Navier - Stokes equations requires to solve three-dimensional Poisson equation. The problem with this equation is enough, because not every method can be useful in solving the Poisson equation with the right hand side term and with complicated boundary conditions.

Moreover in this paper, dimension like 50x50x50 was chosen for test calculations of the cubic computational domain. At the same time mesh each subdomain stored locally on the processor. The calculations of the speedup produced done on a standard PC (Intel (R) Pentium (R) 4 CPU 2.66 GHz, 512 MB RAM), dual-core machine (Intel (R) Core 2 Duo CPU 2.2

GHz, 1 GB RAM) and on URSA cluster (al-Farabi Kazakh National University, Kazakhstan) specifications of equipment listed in the table I below.

Issakhov A. al-Farabi Kazakh National University, Almaty, 050041 Kazakhstan (phone: +7 705 610 80 50; e-mail: aliisahov@mail.ru).

TABLE I
CHARACTERISTICS OF URSA CLUSTER

	URSA
Number of nodes	15 nodes
A node	Intel(R) Xeon(R) CPU E5335 2.00GHz
Memory in each node	10 GB Memory
Network	Gigabit Broadcom NetXtreme II
MPI	MPICH2
System	Scientific Linux 4.6, Oscar 5.0

As an example, we consider the three-dimensional numerical solution of the Poisson equation. It is established that the possibility of experimental studies applied for solving the Poisson equation for calculating the pressure field is very limited. Because of these circumstances, mathematical modeling is particularly important method for solving these kinds of equations. However, the specific tasks and imposes restrict the numerical methods. First of all, this is dictated by economic inefficiency in the number of actions and, mainly, the need of large computer memory resources. There are many methods for solving the Poisson equation - the method of upper relaxation, decomposition method, etc., but they are not effective due to the presence of weight functions. This work proposes the use of the Fourier method [1]-[3] in conjunction with the tridiagonal matrix method (Thomas algorithm), which is used to determine the Fourier coefficients. This algorithm is more accurate as opposed to iterative methods, and also simplifies the process of computing through the use of Fourier transforms, which agrees well with analytical solutions. However, the simulation can be used in two ways. In some cases the problem can be simplified using the boundary conditions of the first kind, while the other approach is to use parallel programming tools of multiprocessor systems [4].

In this connection there arose a problem of the developed parallel algorithm for Fourier method to solve the three-dimensional Poisson equation to calculate the pressure field in the computing clusters. This problem has been resolved with the distribution calculated on a rectangular grid computing subdomains for each processor. The exchange between the calculated values of the required processor is implemented by using MPI library [4]-[8]. The effectiveness of the code was tested on URSA cluster (al-Farabi Kazakh National University, Kazakhstan). During test calculations, the number of processors (up to 10) was obtained by quasi-linear dependence of the speedup on the number of processors.

Parallelization of the Fourier method for solving three-dimensional Poisson equation

Consider the three-dimensional Poisson equation, the complex right-hand side term and not the usual boundary conditions, which can be written as follows:

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}+\frac{\partial^2 p}{\partial z^2}+10\,p=-4\cos(3x+y-2z)+12e^{x-z}+10, \quad (1)$$

With boundary conditions:

$$\frac{\partial p}{\partial z}=-2\sin(3x+y-2z)-\exp(x-z) \text{ – right side,} \quad (2)$$

p=cos(3x+y-2z)+exp(x-z)+1 – other sides.

Here p – pressure, x,y,z – space coordinates

The equation for the pressure (1) is approximated at the point (i, j, k) and takes the following form:

$$\frac{p_{i+1,j,k}-2p_{i,j,k}+p_{i-1,j,k}}{\Delta x^2}+\frac{p_{i,j+1,k}-2p_{i,j,k}+p_{i,j-1,k}}{\Delta y^2}+\frac{p_{i,j,k+1}-2p_{i,j,k}+p_{i,j,k-1}}{\Delta z^2}+10p_{i,j,k}=F_{i,j,k} \quad (3)$$

where $F_{i,j,k}=-4\cos(3i\Delta x+j\Delta y-2k\Delta z)+12e^{i\Delta x-k\Delta z}+10$ (4)

The boundary conditions for the Poisson equation we obtain from (2). To equation (3) we are applying the Fourier method, according to which, for any grid function f(i) there is a decomposition

$$f(i)=\frac{2}{N}\sum_{k=0}^{N}\rho_k\phi_k\cos\frac{\pi ki}{N},\ \mathrm{i}=0,1\ldots,\mathrm{N},$$

where

$$\phi_k=\sum_{i=0}^{N}\rho_i f(i)\cos\frac{\pi ki}{N},\ \mathrm{k}=0,1\ldots,\mathrm{N},$$

$$\rho_i=\begin{cases}1, 1\le i\le N-1\\ 0.5, i=0,N\end{cases}$$

According to the relations we have:

$$p_{i,j,k}=\frac{2}{N_3}\sum_{l=0}^{N_3}\rho_l a_{i,j,l}\cos\frac{\pi kl}{N_3},\ F_{i,j,k}=\frac{2}{N_3}\sum_{l=0}^{N_3}\rho_l b_{i,j,l}\cos\frac{\pi kl}{N_3}, \quad (5)$$

where

$$a_{i,j,l}=\sum_{k=0}^{N_3}\rho_k p_{i,j,k}\cos\frac{\pi kl}{N_3},\ b_{i,j,l}=\sum_{k=0}^{N_3}\rho_k F_{i,j,k}\cos\frac{\pi kl}{N_3}, \quad (6)$$

Substitution (5) to equation (3) and the usage of the formula

$$\cos\frac{\pi(k+1)l}{N_3}+\cos\frac{\pi(k-1)l}{N_3}=2\cos\frac{\pi kl}{N_3}\cos\frac{\pi l}{N_3}$$

Making some simplification, then the resulting expression can be written for a fixed k = l and priory divided to $\frac{2}{N_3}\rho_l a_{i,j,l}\cos\frac{\pi kl}{N_3}$, like:

$$\frac{a_{i+1,j}-2a_{i,j}+a_{i-1,j}}{\Delta x^2}+\frac{a_{i,j+1}-2a_{i,j}+a_{i,j-1}}{\Delta y^2}+\frac{a_{i,j}}{\Delta z^2}\left(2\cos\frac{\pi l}{N_3}-2\right)+10a_{i,j}=b_{i,j} \quad (7)$$

This equation is transformed into the following form:

$$-\frac{a_{i,j-1}}{\Delta y^2}+\left[\left(\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2}-\frac{1}{\Delta z^2}\left(2\cos\frac{\pi l}{N_3}-2\right)-10\right)a_{i,j}-\frac{a_{i+1,j}+a_{i-1,j}}{\Delta x^2}\right]-\frac{a_{i,j+1}}{\Delta y^2}=-b_{i,j}$$

As is known, tridiagonal matrix method (Thomas algorithm) refers to the direct methods for solving differential equations and applied to the equations, which can be written as a system of vector equations [1]-[3]

$$-A_j\overrightarrow{a_{j-1}}+B_j\overrightarrow{a_j}-C_j\overrightarrow{a_{j+1}}=\overrightarrow{F_j},\ \mathrm{i}=1,2\ldots\mathrm{N}_2\text{ -1}, \quad (8)$$

Where matrixes A_j, B_j, C_j and vectors $\overrightarrow{F_j},\overrightarrow{a_j}$ are as follows:

$$A_j=\begin{vmatrix}\frac{1}{\Delta y^2} & & 0\\ & \ddots & \\ 0 & & \frac{1}{\Delta y^2}\end{vmatrix},$$

$$B_j=\begin{vmatrix}\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2}-\frac{1}{\Delta z^2}(2\cos\frac{\pi l}{N_3}-2)-10 & -\frac{1}{\Delta x^2} & & 0\\ -\frac{1}{\Delta x^2} & \ddots & & -\frac{1}{\Delta x^2}\\ 0 & & -\frac{1}{\Delta x^2} & \frac{2}{\Delta x^2}+\frac{2}{\Delta y^2}-\frac{1}{\Delta z^2}(2\cos\frac{\pi l}{N_3}-2)-10\end{vmatrix},$$

$$C_j=\begin{vmatrix}\frac{1}{\Delta y^2} & & 0\\ & \ddots & \\ 0 & & \frac{1}{\Delta y^2}\end{vmatrix},\quad \overrightarrow{a_j}=\begin{vmatrix}a_{0,j}\\ \vdots\\ a_{N_1,j}\end{vmatrix},\quad \overrightarrow{F_j}=-\begin{vmatrix}b_{0,j}\\ \vdots\\ b_{N_1,j}\end{vmatrix}.$$

Tridiagonal matrix method (Thomas algorithm) for solving equation (8) can be written as follows:

$$\alpha_{j+1}=(B_j-A_j\alpha_j)^{-1}C_j,\ \mathrm{j}=1,2..\mathrm{N}_2\text{ -1},\ \alpha_1=B_0^{-1}C_0 \quad (9)$$

$$\overrightarrow{\beta_{j+1}}=(C_j-A_j\alpha_j)^{-1}(\overrightarrow{F_j}+A_j\overrightarrow{\beta_j}),\ \mathrm{j}=1,2..\mathrm{N}_2\text{ -1},\ \overrightarrow{\beta_1}=B_0^{-1}\overrightarrow{F_0} \quad (10)$$

$$-A_j\overrightarrow{a_{j-1}}+B_j\overrightarrow{a_j}-C_j\overrightarrow{a_{j+1}}=\overrightarrow{F_j},\ \mathrm{j}=1,2\ldots\mathrm{N}_2\text{ -1},\ \overrightarrow{a_{N_2}}=\overrightarrow{\beta_{N_2+1}} \quad (11)$$

After finding the coefficients ai,j,k the values of the pressure field can be found from equation (5). Usually fast Fourier transform method is applied to calculate the sum of (5) and (6), which allows us to calculate these amounts, as sums for the O(N lnN) operations, which significantly reduces computing time. Increased productivity is shown in Figures 1 and 2. Different lines in Figure 1 indicate the dimension N of subdomains of the computational domain. Figure 2 shows the efficiency of calculations made on different computers, so the chart third column indicates that the computational time required to solve the problem on a URSA cluster, is minimal.

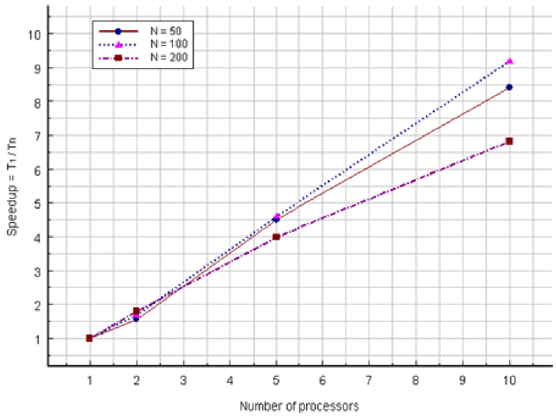

Fig. 1 Calculation of the speedup

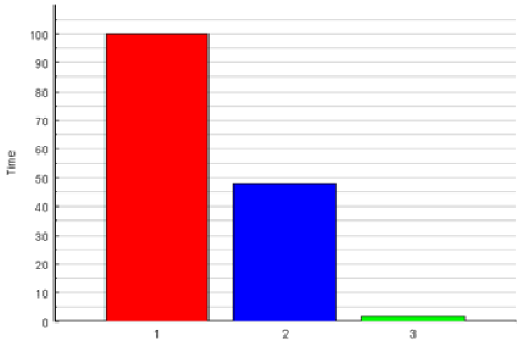

Fig. 2 Calculation of the speedup
1) Intel (R) Pentium (R) 4 CPU 2.66 GHz. 2) Intel (R) Core 2 Duo CPU 2.2 GHz, 1 GB RAM 3) URSA cluster, equipment specifications are given in Table 1

Thus, a new algorithm for solving three-dimensional Poisson equation was constructed within the framework of this paper. Moreover it allows more accurate outcomes as a result of application of OpenMP and MPI technologies of parallel programming, which in its turn significantly reduces computing time and improves the efficiency calculations.

Since the present study belong to a class of fundamental problems, the results can be used to solve complex problems in computational fluid dynamics, which is used to determine the pressure by using Poisson equation.

REFERENCES

[1] V. Ilyin, Numerical analysis. – Novosibirsk: INM and MG , 2004 – p. 71
[2] A. Samarskii, A. Gulin, Numerical methods. – M.: Nauka, 1989. – p. 432
[3] A. Samarskii, Introduction to the theory of difference schemes.–M.: Nauka,1971.– p. 553
[4] Quinn, Michael J., Parallel programming in C with MPI and OpenMP. McGrawHill, 521, 2004.
[5] G. E. Karniadakis, Parallel Scientific Computing in C++ and MPI, 2000.
[6] Pacheco, P. Parallel Programming with MPI. Morgan Kaufmann, 1996.
[7] Alexey Lastoevsky L., Jack Dongarra. High performance heterogeneous computing. Published by John Wiley & Sons, Inc., Hoboken, New Jersey. 269, 2009.
[8] Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta Introduction to Parallel Computing. Second edition. Pearson Education Limited.Edinburgh Gate, Harlow, Essex CM20 2JE, England, 2003.

Alibek Issakhov was born on 1985 May 2 at Almaty, Kazakhstan. I educated in 2008 bachelor degree in Mathematical and computer modeling, al - Farabi Kazakh National university, mechanical mathematical faculty, then in 2010 master degree at same university with same specialization. Currently studying in PhD doctorate in Mathematical and computer modeling at the same university. Have more than 40 publications. Research interests are problems of turbulent flows, mathematical modeling of complex nonlinear physical phenomena, turbulent transport of impurities in a stratified medium, application of mathematical modeling methods for solving problems of environmental protection, computational aspects of environmental problems in the fields of gas and oil, spectral method based on Chebyshev polynomials to model turbulence, method for large eddy simulation for the problem of turbulence, parallel programming in OpenMP and MPI.