The pyramids method application for the synthesis of a parallel algorithm solving finite differential equations scheme Yee. One-dimensional case

L.V. Yablokova[a], D.L. Golovashkin[b], O.V. Kalyuzhnaya[a]

[a] *Samara National Research University, 443086, 34 Moscow highway, Samara, Russia*

[b] *Image Processing Systems Institute of RAS – Branch of the FSRC "Crystallography and Photonics" RAS, 443001,151 Molodogvardeyskaya street, Samara, Russia*

Abstract

In this paper, we propose a modification of the method of the pyramids to build parallel algorithms for explicit difference equations. Its efficiency is shown on a practical example of the differential solution of the one-dimensional equations of the Maxwell in two tasks efficiency. In comparison with a usual parallel algorithm acceleration of calculations is increased by 1.6 times. We investigated the efficiency of the author's approach depending on the height of the pyramid, indicated the limits of applicability of the proposed changes.

Keywords: The method of the pyramids; the grid area; difference solution; parallel algorithm; computing acceleration

1. Introduction

Enhancing the role of mathematical modeling in research is a general scientific trend of the last few decades, a characteristic even for humanitarian branch [1]. In the natural sciences emerged independent directions using computational experiment as the main instrument of the subject area of study: computational Heat Transfer [2], hydrodynamics [3], computer optics [4]. Interests of the authors of this publication are the last of these areas, developing since the mid 80-ies of the last century.

In planning computational experiment, the focus is traditionally given to the construction or the choice of a mathematical model of the phenomenon investigated. In particular, the development of computer optics accompanied by a shift of interest in the scientific community with a beam model (solution of the EIKONAL equation and of the transfer equation) of the wave (Fourier integrals, Fresnel integrals and Kirchhoff integrals) to rigorous diffraction theory (Maxwell's equations). This phenomenon is generally attributed to the improvement of the technological base [5]. It is connected to the transition from micro- to Nano scale inhomogeneity's characteristic optical elements, resulted in new possibilities of optics and new problems that arise in front of her. Indeed, by reducing irregularities specified geometric optics and physical theory of diffraction are no longer satisfactory to describe the process of diffraction and the focus of researchers focuses on the use of the mathematical theory.

However, as a model of the theory [6] and numerical methods relating to them [7], and the possibility of working with Nano-objects (electron microscopy), though less extensive, there is still in the middle of the last century. Making this explanation is not entirely exhaustive. According to the authors of this publication should be added to it the improvement of computer technology, without which the numerical solution of Maxwell's equations is extremely difficult. Unfortunately, this circumstance in the corresponding fundamental literature is either ignored altogether, or is mentioned in passing, as something catches your eye. For example, only in the third edition of the Alan Taflove's monograph [8], who gave the name of the FDTD-method, referred to its implementation on the GPU. Among the plurality of books on the same method, only one [9] and it is not entirely devoted to the peculiarities of the organization of parallel computing. This is despite the explosive profusion of texts in periodicals related to the specific implementations of the difference solution of Maxwell's equations on different processor architectures. This discrepancy shows the urgent need to systematize the extensive experience of the FDTD-method, followed by an indication of the "missing" options for its development.

Without attempting in this paper to solve the first problem, the authors dwell on the second, referring to the use of the pyramid method to the difference solution of Maxwell's equations. Characteristic of this problem as "missing" is connected to the output of fashion algorithms for multiprocessor cluster computing systems and massive circulation of developers to heterogeneous systems with GPUs. However, the known important constraint on implementation [10] of software FDTD-method on GPU wide practices. Specifically, a small amount of video memory compared to RAM of one node in the cluster. For example, the cluster K-100, Institute of applied mathematics, the difference reaches two orders of magnitude [11]. In addition, with the possible development of fog computing [12], while existing only as a theoretical concept, not involving work with video processors, researchers' attention again switches to the peculiarities of organization of arithmetic operations and communications between processors connected by a network.

2. The Maxwell's equations in the one-dimensional case

Choosing one-dimensional case, we mean that it recorded the Maxwell's equations have analytical solution and the case is of interest only to illustrate the proposed hike further to the compilation of parallel algorithms.

For TEM-waves [8] in the free space Maxwell's equations are:

$$\mu_0 \frac{\partial H_y}{\partial t} = -\frac{\partial E_x}{\partial z}, \varepsilon_0 \frac{\partial E_x}{\partial t} = -\frac{\partial H_y}{\partial z} \quad (1)$$

Where μ_0 and ε_0 - electric and magnetic constants, axis OZ is the direction of propagation, E_x and H_y the projection of the electric and magnetic fields on the respective axes. At the boundaries of the field of computational experiment $D = \{(t,z): 0 \le t \le T; 0 \le z \le L\}$ on function E_x impose Dirichlet conditions: $E_x(t,0) = 0, E_x(t,L) = 0$ appropriate electrical wall. As the initial conditions are field missing, take the electric and magnetic fields at $t = 0$. Radiation in the region D is administered by "hard" [8] source $E_x(t,z_s) = \mathrm{Re}\left\{e^{-i\omega t+\tilde{z}}\right\}$. Where ω is cyclic frequency, z_s is the place location of the radiation source.

Difference solution (1) is accompanied on the overlay D grid area D^h, at the nodes $\left\{(t_n, z_k): t_n = nh_t, n = 0,1,...N = T/h_t, k = 0,...,K = L/h_z\right\}$ where the function is defined grid $E^n_{x_k}$, and the nodes $\left\{(t_{n+0.5}, z_{k+0.5}): t_n = nh_t, n = -1,0,1,...N-1, k = 0,...,K-1\right\}$ is defined grid $H^{n+0.5}_{x_{k+0.5}}$. The obvious difference scheme Yee [8] for (1) is traditionally written as:

$$\mu_0 \frac{H^{n+0.5}_{y_{k+0.5}} - H^{n-0.5}_{y_{k+0.5}}}{h_t} = -\frac{E^n_{x_{k+1}} - E^n_{x_k}}{h_z}, \varepsilon_0 \frac{E^{n+1}_{x_k} - E^n_{x_k}}{h_t} = -\frac{H^{n+0.5}_{y_{k+0.5}} - H^{n+0.5}_{y_{k-0.5}}}{h_z} \quad (2)$$

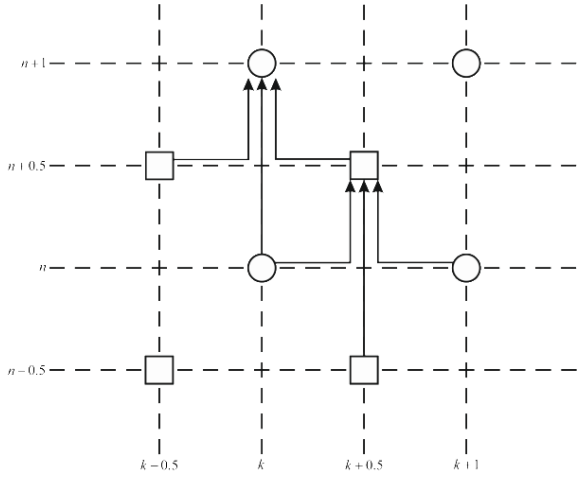

Fig. 1. The differential template for the scheme (2). Circumference depicted components for the electric field, squares depicted components for the magnetic.

Separate arrangement of different nodes grid functions (Fig. 1) provides higher-order approximation scheme of equations (1) with respect to time and space, allows you to not worry about setting the boundary conditions for the magnetic field.

3. The classic pyramids method

Pyramids method proposed and developed in [13, 14] can be carried to the theory of automatic parallelization cyclic fragments of sequential programs. Compared to other approaches to solve this problem, for instance the methods of hyper planes, coordinates and parallelepipeds, it allows you to work with cycles, the parameters of which are included in the non-linear expressions. The disadvantage of the method is considered to be impossible to use for simple cycles.

The cyclic structure of the general form, with any number of nested loops, put in bijective correspondence space iterations, each vector is associated with single design iteration. Entered binary dependency relationships and following in the space [15]. The first step of the method is accompanied by the release of the space of iterations resulting vectors, each of which does not depend on any other vector space. In the second step to a problem parallel algorithm include a resultant vector of all vectors of the space on which it depends. Building a parallel algorithm completes ordering vectors in each task in accordance with the following relationship. A distinctive feature of this algorithm is a complete lack of communication between its objectives.

For the scheme (1) space iterations obviously coincide with the grid area D^h except nodes, which define the initial and boundary conditions. Grid functions initial and boundary conditions are known and cannot be calculated. The ratio is determined depending on the immediate differential circuit pattern (Fig. 1) and associates the vector $(n+1,k)$ with (n,k), $(n+0.5,k-0.5)$ $(n+0.5,k+0.5)$ vectors and the vector $(n+0.5,k+0.5)$ with $(n-0.5,k+0.5)$, (n,k), $(n,k+1)$ vectors. A ratio of the dependency is constructed as a transitive closure of the immediate relationship of dependence on space of iterations. A ratio of the sequence is determined by the order of traversal of the grid region in the production of computing serial algorithm. Namely, the outer loop iterates over the layers of time in the direction of increasing values of the index n, and the internal nodes iterates over the layers of space in the direction of increasing k. At integer values of n are grid values of the electric field at non-integer values of n are the discrete values of the magnetic field.

The resulting vectors will correspond to nodes with $n = N, k = 1,...,K-1$. Let the task μ of parallel algorithm classified the resulting vector (N,μ). Then on the layer n (general) the selected task will be to calculate the grid function at the nodes k from $\mu - N + n$ to $\mu + N - n$ and the layer $n+0.5$ will be to calculate the grid function at the nodes k from $\mu - N + n + 0.5$ to

$\mu + N - n - 0.5$. It is easy to see (Fig. 2), which is grid region the nodes within the jurisdiction of the μ task will be inside the triangle (in two-dimensional case, inside the pyramid).

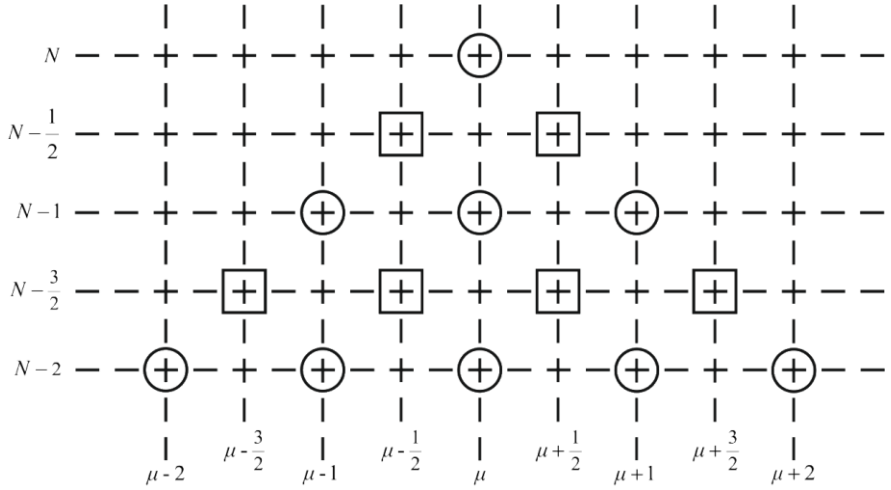

Fig. 2. Vectors-space iterations within the jurisdiction of the μ task parallel algorithm.

We have gotten rid of communications due to duplication of arithmetic operations in a variety of tasks. Therefore, the task $\mu - 1$ contains the n layer are in common with the μ task of vector $k = \mu - N + n ... \mu + N - n - 1$. Therefore, that duplicated all their vectors other than two nodes. And if the communication costs in the conventional parallel algorithm [9] with increasing N increases linearly (in each time layer several parcels / receptions for the problem), the volume of duplicate arithmetic operations using the pyramid method increases quadratic (in proportion to the area of a triangle) that it causes the failure of the method for real values of N.

4. A modification of the method of the pyramids

To overcome this drawback, the authors have developed the following modification of the method of the pyramids. Initially, we restrict consideration to h (the height of the pyramid in integer layers) layers of time instead of N. Take to doing one task ξ adjacent to each other of the resulting vector on layer $h+1$, thus obtaining K/ξ task parallel algorithm. Now we will build up over to the resulting μ $(1 < \mu < K/\xi)$ task, an isosceles trapezoid (instead of a pyramid) with the top base of ξ and the bottom base of $\xi + 2h$ vectors are the same of the trapezoid until reaching the N layers.

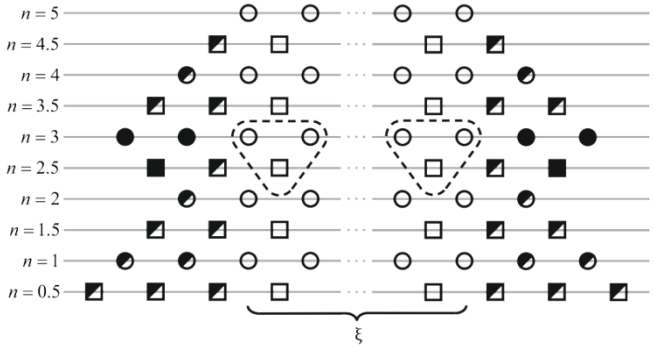

Fig. 3. Vectors space iterations within the jurisdiction of μ problem parallel algorithm in a modified method of the pyramids. Example h = 2 and N = 5.

In Fig. 3, illustrating the proposed modification, half-shaded vector space of iterations does not belong to the μ task in a traditional parallel algorithm [9]. The vectors listed in the left of figure treated exclusively to the problem of $\mu - 1$ vectors indicated on the right of the task concerned solely to $\mu + 1$. Now calculate them will be duplicated. In his turn, the adjacent task will be to duplicate the calculation of the net functions not related to a filled grid points. Values of network functions in the nodes, the filled completely formed μ task, but it is needed for further calculations on the layers $k = 3.5$ and $k = 4$, and on this subject received from the $\mu - 1$ task (shaded nodes on the left) and $\mu + 1$ (shaded nodes on the right). In his turn, neighboring problems should be referred to the values of the functions in the circled nodes: left trace passes $\mu + 1$ task, right trace passes $\mu + 1$ task. Now, communication between tasks of a parallel algorithm is not to make at each time layer as in the conventional parallel algorithm [9], but not existent, as in the classic method pyramids. They are implemented using integer h layers. When $h = 1$, we get the usual parallel algorithm, with $h = N$, we get of algorithm built using the traditional method of pyramids. A distinctive feature of the new approach is the introduction of additional consideration on the space of iterations varied parameter. The parameter is the height of the pyramid, changing which can be expecting to achieve a minimum duration for the calculation (2).

5. Experimental study of the modified method

The purpose of the pilot study was to demonstrate the feasibility of accelerating the computation by the parallel algorithm, compiled with the help of a modified method of the pyramids.

As the hardware base is selected a dual-processor computer (AMD Opteron 246), that has allowed excluding the mutual influence of computational threads running on a single core or on different cores located in a single chip. A computer running a modern operating system Debian 8.5, was used GFortran 4.9.2, and a library OpenMPI 1.6.5. The authors recognize that the organization of parallel computing on shared memory, it is customary to refer to the Open Multi-Processing standard, but in the future we plan to compare with the package MEEP, which is written using MPI. The choice of the programming language Fortran is caused to the convenience of the vector notation, the use of which will reduce the amount of cyclic structures in the program. So, we calculate for one temporal layer is shaped in the form of vector operations without iterating through the nodes in space in a loop. It can be done from common programming languages, only Matlab, but with a significant increase in the duration calculation.

Parameters of numerical experiments (the number of nodes in the grid region on the space $K = 500$, the number of nodes in the grid region at the time $N = 756000$, the number of tasks of parallel algorithm 2, and the height h of the pyramids vary from 2 to 200) were selected from following considerations:

- the steps of grid region satisfy the Courant condition of stability of difference scheme (2);
- the duration of numerical experiments is sufficient to eliminate the influence of short-term system events(tab. 1, the second and third columns);
- the number of layers at a time evenly divides the height of all selected to study the pyramids (tab. 1, first column) and sufficient to obtain consistent estimates of the mathematical expectation of acceleration (tab. 1, last column);
- the number of nodes in space allows you to work with pyramid maximum height (tab. 1, last line: $200 < {}^{500}\!/_{2}$).

The duration calculation for a sequential program made up 3.2890 sec., according to the traditional parallel (author's implementation of the algorithm from [9]) 3.3085 sec., i.e. acceleration is not achieved. The use of the classical method of the pyramids is impossible ($N >> K$).

Table 1. A The results of computational experiments

the height of the pyramid h	the duration of calculation without unfolding the cycles (sec.)	the duration of this computation with unfolding the cycles (sec.)	acceleration
2	3.8593	2.5742	1.2776
3	3.6640	2.4296	1.3536
4	3.5546	2.2773	1.4442
5	3.4804	2.2031	1.4929
6	3.4375	2.1992	1.4955
7	3.4257	2.1484	1.5309
8	3.4140	2.0898	1.5738
9	3.3867	2.0898	1.5738
10	3.3632	2.0820	1.5797
20	3.3515	2.0312	1.6192
30	3.4023	2.0390	1.6130
40	3.4296	2.0625	1.5946
50	3.5039	2.0898	1.5738
60	3.5468	2.1484	1.5309
70	3.5976	2.1835	1.5062
80	3.6601	2.1875	1.5035
90	3.7226	2.1914	1.5008
100	3.7773	2.2304	1.4746
200	4.3906	2.6054	1.2623

The software implementation of the modified method of the pyramids with the introduction of nested loop for the production of calculations within one pyramid (the step of the outer loop iterates over the layers at the same time were set equal to h) were not successful (tab. 1, second column), the duration of calculations has increased significantly. The authors suggested that this was due to the additional looping constructs and resorted to “expand” nested loop, which wrote the new program in the Matlab that generates code for the main parallel program.

The authors guess and the performance of the said acceptance was confirmed in the course of numerical experiments (tab. 1, the third and fourth columns). For the pyramid with a height of 20 knots grid region achieved an acceleration of 1.6 times

compared to the sequential implementation. The dependence of the duration of the calculation of the height of the pyramid has a U shaped appearance, due to the nature of the changes in the share of communication and arithmetic operations in the overall computation time with the growth of h. So, for a small elevation gain from the reduction of communication (in h times compared with a conventional parallel algorithm) obviously exceeds the loss of time to doubling of computing the grid functions in General for the task nodes in the grid region (Fig. 3, half shaded nodes), the number of which is small. With the linear growth of h, the number of total nodes grows quadratically for $h > 20$ the amount of additional compared with a conventional parallel algorithm of arithmetic operations is beginning to have a decisive influence on the total duration of the calculation, which is due to this fact increased despite further reductions in communication costs.

6. Conclusion

The use of the author modifications of the classical method of differential pyramids for the solution of Maxwell's equations in one-dimensional case allowed for the selected example to demonstrate the effectiveness of the proposed approach. This suggests the success of further development of the method for the case of a larger number of processors and for multidimensional problems of mathematical physics.

As the field of application of the developed technique of compiling parallel algorithms, the authors identify cases for which a high proportion of communication costs in the total duration of the calculations by the usual parallel algorithm. Otherwise, the use of the modified method of the pyramids, not only will not lead to growth acceleration by reducing these costs, but also will lead to his downfall because of the increase of the duration of the arithmetic operations associated with the method.

Acknowledgements

The research leading to these results has received funding from the Russian Science Foundation grant №16-47-630560-r_a.

References

[1] Vorobyev, V. Mathematical models of historical processes / V. Vorobyev, Y. Berezovskaya // Applied discrete mathematics – 2012. – Vol. 16(2). – P. 105-125.

[2] Samarskii, A. Computational heat transfer / A. Samarskii, P. Vabishchevich – M: Editorial URSS, 2003. – 784 p.

[3] Wesseling, P. Principles of Computational Fluid Dynamics / P. Wesseling – Springer, 2009. – 606 p.

[4] Access mode http://www.computeroptics.smr.ru/

[5] Klimov, V. Nanoplasmonics / V. Klimov – M.: Fizmatlit, 2009. – 480 p.

[6] Born, M. Principles of optics. Electromagnetic theory of propagation, interference and diffraction of light / M.Born, E. Wolf – London: Pergamon Press, 1959.

[7] Cron, G. Equivalent circuit of the field equations of Maxwell, I Proc. IRE, 1944. 32. 289–299.

[8] Taflove, A. Computational Electrodynamics: The Finite-Difference Time-Domain Method / A. Taflove, S. Hagness. – 3-th ed. – Boston: Arthech House Publishers, 2005. – 1006 p.

[9] Wenhua, Yu, Raj Mittra, Tao Su, Yongjun Liu, Xiaoling Yang Parallel Finite-Difference Time-Domain Method / Yu Wenhua, Raj Mittra, Tao Su, Yongjun Liu, Xiaoling Yang Boston: Arthech House Publishers, 2005. – 272 p.

[10] Wahl, P. B-CALM: An open-source GPU-based 3D-FDTD with multi-pole dispersion for plasmonics / P. Wahl, D.-S. Ly-Gagnon, Ch. Debaes, D.A.B. Miller, H. Thienpont // Optical and Quantum Electronics. – 2012. – Vol. 44. – P. 285-290.

[11] Access mode http://www.kiam.ru/MVS/resourses/k100.html

[12] Bar-Magen Numhauser, Jonathan (2013). Fog Computing introduction to a New Cloud Evolution. Escrituras silenciadas: paisaje como historiografía. Spain: University of Alcala. P. 111–126.

[13] Lamport, L. The coordinate method for the parallel execution of DO-loops /L. Lamport // Sagamore Computer Conf. on Parallel Proc. IEEE, New York, 1973. P. 1–12.

[14] Valkovsky,V. The Parallel execution of loops. The method of the pyramids /V. Valkovsky // Cybernetics. – 1983.Vol 5.P. 51-55.

[15] Valkovsky,V Elements of parallel programming / V.Valkovsky,V. Kotov, A. Marchuk, N.Mirenkov .– M.: Radio and communication, 1983.–239 p.