On Fast Multipole Algorithms of Velocity Computation in Vortex Element Method for 3D Flows

Ilia K. Marchevsky
iliamarchevsky@mail.ru

Abstract

Some problems of computation acceleration in vortex element method are investigated for the model problem of spatial vortex structures dynamics simulation in inviscid incompressible flow. The so-called 'Symmetrical Vortex Fragmenton' model is used as the vortex element model, and it allows to simulate such phenomena as vortex lines stretch and reconnection. There are several possible ways to computations acceleration: the original algorithm parallelizing with MPI and/or OpenMP technologies, fast approximate methods usage based on the multipole expansion of the velocity field, as well as a combination of these approaches.

1 Introduction

Numerical simulation of 2D and 3D flows and unsteady aerodynamic loads computation is a very complicated problem. In case of external flows with low subsonic velocities, meshless Lagrangian methods (discrete vortex method, vortex blobs/vortex particles method, vortex element method) sometimes are the most efficient from computational point of view. The main value in these methods is vorticity carried by vortex elements (VEs), while velocity and pressure are computed using Biot — Savart law and Cauchy — Lagrange integral analogue. VEs motion equations solution is based on VEs pair interactions computation, which makes this problem similar to gravitational N-body problem. If we need to increase the accuracy of simulation, number of VEs should be increased while computational complexity increases proportionally to the square of VEs number.

There are two main approaches to computations acceleration in vortex element method. The first approach is connected with parallel algorithms usage; for example, it allows to reduce the computation time by 9-10 times for 16-core cluster. Computation time remains acceptable for cases when VEs number doesn't exceed 10-20 thousand for 3D and rather more for 2D problems. However, to improve the accuracy of simulation it is necessary to increase VEs number while computation time should not increase significantly.

In order to achieve this aim we can use the second approach which assumes usage of fast multipole algorithms, similar to ones for N-body problem. An effective fast multipole method for the approximate solution of N-body problem was proposed by J.Barnes & P.Hut [1], its modification for 2D vortex element method developed by G.Ya.Dynnikova [2]. This method is based on a tree construction and approximate calculation of the influence of VEs located in tree cells, far enough away from each other. Computational complexity of this method is proportional to $N \lg N$ against N^2 for the direct method. Fast multipole method usage allows to increase VEs number to several million. It should be noted that computational cost of one operation of VEs influence calculation in 3D case

is approximately 100 times bigger than in 2D case, so fast multipole method developing for 3D flows is an actual problem.

2 Vortex element method with Symmetrical Vortex Fragmentons

Vortex element method is well-known [3, 4], and it allows to solve complicated unsteady problems of aeroelastic fluid-structure interaction. It should be noted that this meshfree method is very attractive because of sufficiently low computational cost in comparison with other CFD methods.

Vortex element (VE) is an element of vorticity field, which generates velocity field in the whole space. Total vorticity field is the superposition of elementary fields generated by certain vortex elements. There are some different types of vortex elements, for example, vortex filament, vortex ring, Novikov's vorton [5], vortex blob etc., each type has both some advantages and limitations. One of the main questions arising when solving a particular problem is the vortex element type choice.

In this research some new type of VE is used — Symmetrical Vortex Fragmenton [6]. Each vortex fragmenton is described by its marker position $\underline{r}_0$, fragmenton vector $\underline{h}$ and intensity Γ (Fig. 1).

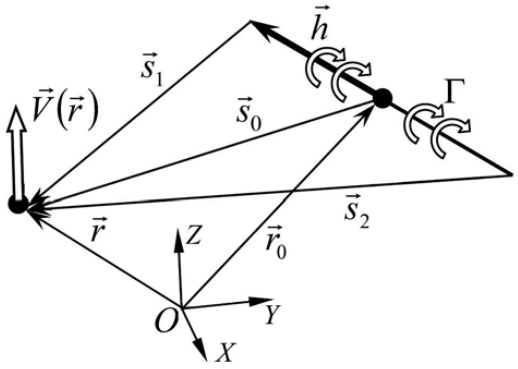

Figure 1: Symmetrical vortex fragmenton model

Main advantages of vortex fragmentons are the following: 1) closed vortex framework can be constructed from vortex fragmentons of the equal intensity; 2) vortex structures in the flow can be simulated with sufficiently small number of vortex fragmentons in comparison with Novikov's vortons and vortex blobs; 3) vorticity field line reconnection process can be simulated with vortex fragmentons. For symmetrical vortex fragmenton analytical formulae for vector potential, velocity and vorticity fields are obtained:

$$\begin{aligned}
\underline{\Psi}(\underline{r}) &= \frac{\Gamma \underline{e}}{4\pi} \ln \frac{|\underline{s}_1| - \underline{e} \cdot \underline{s}_1}{|\underline{s}_2| - \underline{e} \cdot \underline{s}_2}, \\
\underline{V}(\underline{r}) &= \operatorname{curl} \underline{\Psi} = \frac{\Gamma}{4\pi} \left(\frac{\underline{s}_1 - \underline{e} \cdot |\underline{s}_1|}{|\underline{s}_1|(|\underline{s}_1| - \underline{e} \cdot \underline{s}_1)} - \frac{\underline{s}_2 - \underline{e} \cdot |\underline{s}_2|}{|\underline{s}_2|(|\underline{s}_2| - \underline{e} \cdot \underline{s}_2)} \right) \times \underline{e}, \\
\underline{\Omega}(\underline{r}) &= \operatorname{curl} \underline{V} = \frac{\Gamma}{4\pi} \left(\frac{\underline{s}_1}{|\underline{s}_1|^3} - \frac{\underline{s}_2}{|\underline{s}_2|^3} \right).
\end{aligned} \tag{1}$$

In (1) we noted

$$h = |\underline{h}|, \qquad \underline{e} = \frac{\underline{h}}{h}, \qquad \underline{s}_1 = \underline{r} - \underline{r}_0 - \underline{h}, \qquad \underline{s}_2 = \underline{r} - \underline{r}_0 + \underline{h}.$$

Symmetrical vortex fragmenton can be considered as generalization of Novikov's vorton. If $\underline{\Psi}_0(\underline{r})$, $\underline{V}_0(\underline{r})$ and $\underline{\Omega}_0(\underline{r})$ are correspondingly vector potential, velocity and vorticity fields generated by Novikov's vorton of unit intensity and placed in the origin of coordinates, for symmetrical vortex fragmenton we obtain

$$\underline{\Psi}(\underline{r}) = \Gamma \int_{-1}^{1} \underline{\Psi}_0(\underline{r} - \underline{h}t)\, dt, \quad \underline{V}(\underline{r}) = \Gamma \int_{-1}^{1} \underline{V}_0(\underline{r} - \underline{h}t)\, dt,$$
$$\underline{\Omega}(\underline{r}) = \Gamma \int_{-1}^{1} \underline{\Omega}_0(\underline{r} - \underline{h}t)\, dt.$$

These equations allow to use number of results obtained by Alkemade [7] for Novikov's vorton. For example, it can be easily shown that vorticity field generated by symmetrical vortex fragmenton is divergence-free.

3 Vorticity & velocity fields simulation

Total voricity and velocity fields are considered as superpositions of elementary fields generated by N separate vortex fragmentons:

$$\underline{V}(\underline{r}) = \sum_{i=1}^{N} \underline{V}_i(\underline{r}), \qquad \underline{\Omega}(\underline{r}) = \sum_{i=1}^{N} \underline{\Omega}_i(\underline{r}).$$

Because of i-th vortex fragmenton is described by marker position $\underline{r}_{0i}$, fragmenton vector $\underline{h}_i$ and intensity Γ_i (Fig. 1), velocity and vorticity fields evolution could be considered via these parameters variation in time. Fragmenton intensities are supposed to be constants; fragmenton markers are moving with mean value of flow velocity along the fragmenton; their vectors should satisfy Helmholtz equation:

$$\begin{aligned} &\frac{d\Gamma_i}{dt} = 0, \\ &\frac{d\underline{r}_{0i}}{dt} = \frac{1}{2h_i} \int_{-1}^{1} \underline{V}(\underline{r}_{0i} + \underline{h}_i \xi) d\xi, \\ &\frac{d\underline{h}_i}{dt} = \Big(\frac{1}{2h_i} \int_{-1}^{1} \hat{H}(\underline{r}_{0i} + \underline{h}_i \xi) d\xi \Big) \cdot \underline{h}_i. \end{aligned} \tag{2}$$

Tensor $\hat{H}$ could be calculated using different numerical schemes:

- 'direct' scheme: $\hat{H} = \hat{B}$;
- 'transposed' scheme: $\hat{H} = \hat{B}^{\mathrm{T}}$;
- 'symmetrical' scheme: $\hat{H} = \frac{1}{2}(\hat{B} + \hat{B}^{\mathrm{T}})$,

where $\hat{B}(\underline{r}) = \mathrm{grad}\underline{V}(\underline{r})$, $\big(\hat{B}(\underline{r})\big)_{ij} = \dfrac{\partial V_i(\underline{r})}{\partial x_j}$ could be calculated analytically from (1).

Integrals in right side in equations (2) could be calculated numerically using Gaussian quadratures and the whole ODE system (2) could be solved numerically, for example, using Runge — Kutta explicit method.

4 The model problem description

In [8] for the problem of vortex rings momement simulation it was shown that the numerical scheme with symmetric vortex fragmentons allows to obtain strictly periodical solution for more computing time compared to other types of VEs. This indicates a high precision of this scheme when vortex lines stretching simulation.

In this paper, the problem of vortex rings leapfrogging phenomenon simulation is used to test the proposed methods for calculations acceleration. As an initial vorticity field $\underline{\Omega}(\underline{r}, t_0)$ for $t_0 = 0$, we consider two toroidal vortex rings with the same radius $R_1 = R_2 = 1$, which cross sections are circular with a radius of $r_R = 0{,}1$ and which are spaced at a distance of $z = 1{,}2$ as shown in Fig. 2.

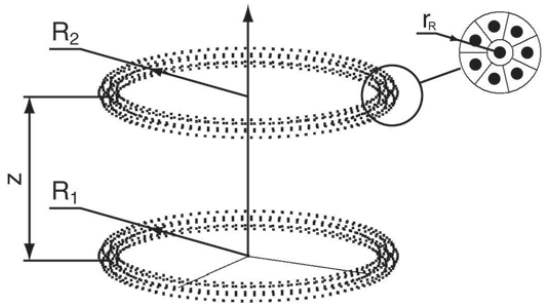

Figure 2: The design scheme for vortex rings leapfrogging phenomenon simulation

Vorticity distribution in the cross section of the vortex ring is the following:

$$|\Omega_0| = A\,(r_R - r)\,/r_R,$$

where A is a constant which can be chosen from the condition of the unit velocity circulation along a contour surrounding the section of the torus. Approximation of the vorticity in the torus with N_V vortex elements was produced by decomposition of the torus into cells of approximately equal size (Fig. 3) and replacement the vorticity in every cell with equivalent circulation VE.

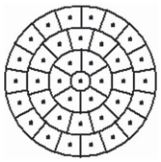

Figure 3: Splitting the cross-section of the vortex ring into the cell

Numerical solution of the ODE system (2) is implemented by the Runge — Kutta method of the second accuracy order with a time step $\Delta t = 0{,}005$. Vortex element smoothing radius was assumed to be $\varepsilon = 0{,}10$.

5 Fast multipole method usage

The first step of the fast multipole method algorithm for calculating the velocity of VEs is a hierarchical tree structure constructing. Each cell of the tree is a parallelepiped which includes some VEs (Fig. 4).

The zero level cell contains all the VEs. It is divided by the longest edge into two identical cells of the first level. All VEs are checked against their belonging to one of this

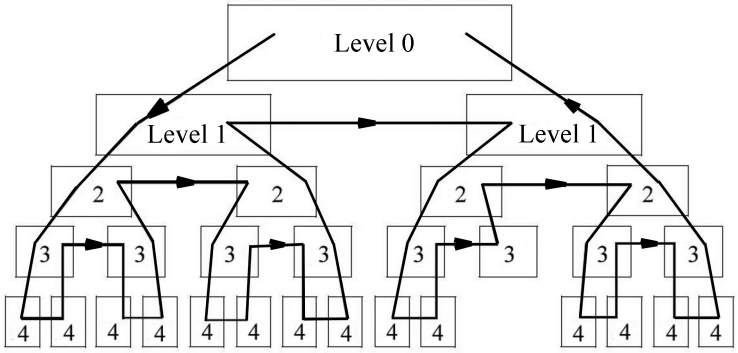

Figure 4: The tree which has 4 levels maximum depth, and the direction of the tree traverse

cells. After that, each cell is 'clipped' on all dimensions to exclude the space without VEs. Note that such a simple procedure allows on the one hand, to increase the efficiency of the method, on the other — to simplify the algorithm.

Next, in a similar way these parallelepipeds are divided in half, forming cells of the second level. Division stops when the specified criterion of cell size, number of VEs in the cell and/or the number of levels is achieved. Note that 'clipping' procedure mentioned above provides all parallelepipeds to be not empty.

On the second step for each tree cell position of the equivalent vortex fragmenton and its intensity vector should be defined, as well as its intensity. Equivalent VE position and vector are calculated in two steps: firstly the average position and average vortex fragmenton vector are computed:

$$\underline{\xi}_e = \frac{\sum \underline{r}_i |\Gamma_i|}{\sum |\Gamma_i|}, \quad \underline{\eta}_e = \frac{\sum \underline{h}_i |\Gamma_i|}{\sum |\Gamma_i|},$$

where the summation is over all the VEs, located in the current cell of the tree.

The beginning and end of the equivalent VE are the points of intersection of the line passing through the point $\underline{\xi}_e$ in the direction of $\underline{\eta}_e$, with the faces of the cell. The center of this line determines the position $\underline{r}_e$ of the equivalent VE, while its vector $\underline{h}_e$ is collinear with the vector $\underline{\eta}_e$, and its length is equal to half the length of the equivalent VE (Fig. 5).

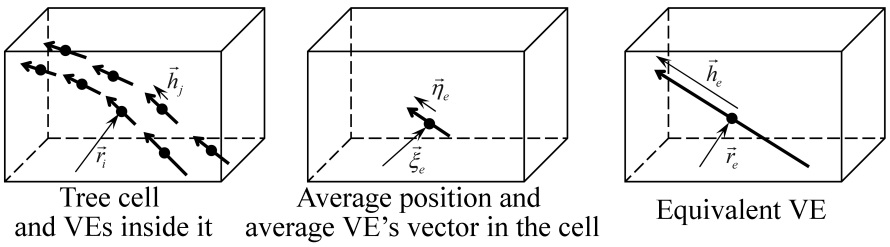

Figure 5: Algorithm for equivalent vortex fragmenton computing

The intensity of the equivalent vortex fragmenton is calculated using the following formula:

$$\Gamma_e = \frac{|\underline{\eta}_e|}{|\underline{h}_e|} \cdot \sum \Gamma_i.$$

It should be noted that for the vertices of the tree (i.e. for the cells which don't have child cells on the next level), these values are calculated directly. In addition the sum

of the absolute values of intensities of VEs should be calculated and stored and also the so-called 'moments':

$$\Gamma_e^* = \frac{|\underline{\eta}_e|}{|\underline{h}_e|} \cdot \sum |\Gamma_i| \,, \quad \underline{M}_e^r = \sum \underline{r}_i \, |\Gamma_i| \,, \quad \underline{M}_e^h = \sum \underline{h}_i \, |\Gamma_i| \,.$$

For cells on the higher levels 'moments', total intensity and the total sum of the absolute values of intensities of VEs are calculated by summation of the corresponding characteristics of child cells, so the average position and the average VE vector will be determined by the formulae

$$\underline{\xi}_e = \frac{\underline{M}_e^r}{\Gamma_e^*}, \quad \underline{\eta}_e = \frac{\underline{M}_e^h}{\Gamma_e^*}.$$

It is important to note that the procedure of the tree construction and its cells characteristics computation is extremely low-cost, and the numerical experiment shows that the time for it increases almost linearly with the number of VEs. In the Table 15 the ratio of the time of tree construction to the time of vortex-to-vortex pair influences computation is shown for different numbers of VEs in the flow when vortex-to-vortex influences are competed using 'direct' method.

N_V	2 000	4 000	8 000	12 000	16 000	20 000
$\frac{t_{tree}}{t_{slow}} \cdot 100\,\%$	0,130	0,049	0,025	0,023	0,020	0,015

Table 15: Time needed for the tree construction referred to the time of problem solving using 'direct' method (in %)

Thus, the tree constructing which is difficult for parallelization, isn't a 'bottleneck' of the fast multipole method algorithm. It should also be noted that in software implementation of this procedure dynamically allocated memory is widely used, so when working with clusters with distributed memory procedure of tree broadcasting to all compute nodes is non-trivial. In fact, the most efficient way to overcome this problem is broadcasting of all VE's parameters to all compute nodes and independent trees building on all nodes. Due to the complete determination of the algorithm and identity of the source data, all the trees will be identical.

On Fig. 6 tree cells (parallelepipeds) of the $1 \ldots 8$ levels are shown for the problem of two vortex rings leapfrogging simulation.

On the third step velocities are calculated for VEs in tree vertices. The influence of VEs from the same tree vertex as well as influence of VEs from tree vertices placed close to the selected vertex is calculated directly using Biot — Savart law. Then tree traverse is carried out and the influence of VEs located in far tree cells is calculated approximately as influence of the equivalent vortex fragmenton.

On Fig. 7 it is shown how the influence on the selected VE (indicated by a circle) is calculated from all other vortex fragmentons which form vortex rings.

A direct calculation of influence takes place only for sufficiently small number of VEs, which are close to the selected one, while for the parts of the tree which are shown on Fig. 7 as parallelepipeds, influence is calculated as from the equivalent vortex fragmentons.

The criterion of proximity of two cells is the ratio of the sum of lengths of their diagonals to the distance between positions of their equivalent vortex fragmentons. On Fig. 7 the criterion of proximity is the following: two tree cells are considered to be close one to the

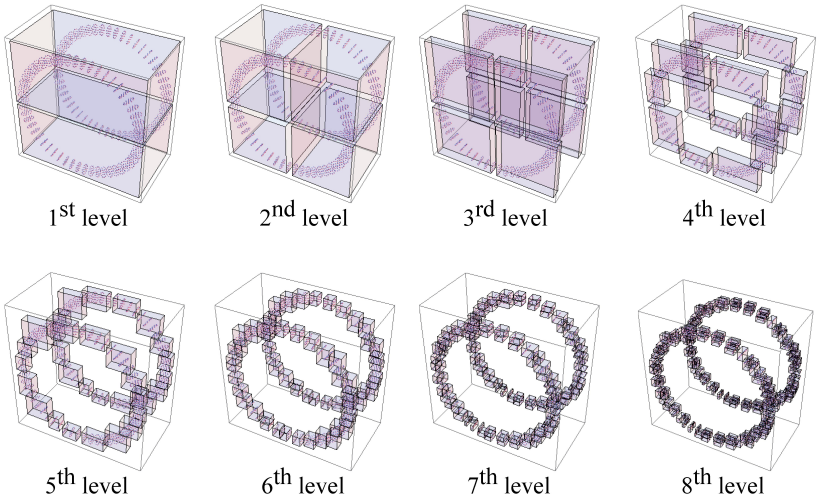

Figure 6: Tree cells of 1 . . . 8 levels

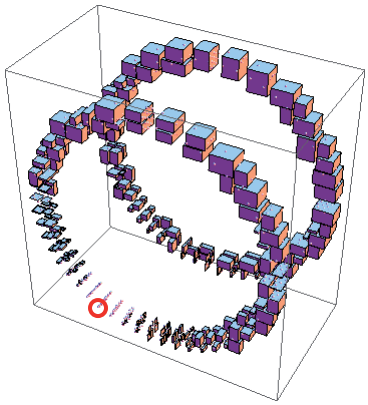

Figure 7: A scheme for vortex influence computing on the selected VE (indicated by a circle)

other if the specified ratio more then $\theta = 0{,}25$. On Fig. 8 schemes of vortex influence computation are shown for the same problem when $\theta = 0{,}10$ and $\theta = 0{,}50$.

It is clear that with increasing the value of θ computation speed increases, but the accuracy of the results reduces. The value of $\theta = 0$ formally means that the proximity criterion is always satisfied, so the fast multipole method algorithm doesn't work and the calculation is made using the direct method. The computation time dependence on the value of θ for numerical schemes with 2 000, 4 000, 8 000 and 20 000 VEs is shown on Fig. 9 (all results are referred to the computation time for corresponding problem solving using the direct method). It can be seen that the higher computational complexity of the problem is, the more effective fast multipole algorithm becomes.

It should be noted that for a fixed value of θ (i.e. for the fixed error level) computation speed greatly depends on the maximum number of levels of the tree, which is given in the algorithm for its construction. The error at 2-3 levels from the optimum value leads to a very significant increase in computation time.

Calculation shows that in order to achieve the accuracy which is proper for practical purposes, the value of θ should be chosen not more then $0{,}2 \ldots 0{,}3$. Further in all calculations we consider $\theta = 0{,}2$.

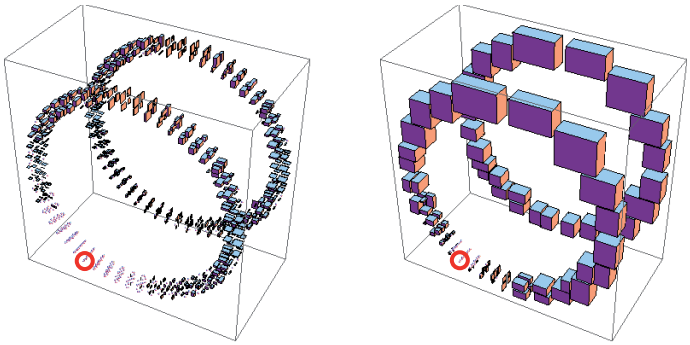

Figure 8: Schemes for vortex influence computing on the selected VE (indicated by a circle) when $\theta = 0{,}10$ (left graph) and $\theta = 0{,}50$ (right graph)

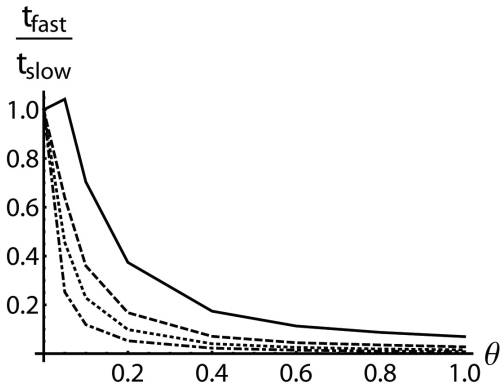

Figure 9: Copmutational time dependence on the value of θ (solid line – 2 000 VEs, dotted line – 4 000 VEs, points – 8 000 VEs, dash-dotted line – 20 000 VEs)

6 Parallel computational algorithms usage

The fast multipole method usage for vortex fragmentons velocities computation allows to reduce the computation time significantly. In Table 16 the ratio is given of the run-time of one step of the calculation when using fast multipole and direct methods when number of VEs in the numerical scheme is sufficiently large.

N_V	10 000	20 000	30 000	40 000	50 000
$\frac{t_{fast}}{t_{slow}}$	0,090	0,053	0,033	0,029	0,022

Table 16: The ratio of computation time for fast multipole and direct methods

It can be seen that computation speed for numerical schemes which contain tens of thousands of VEs (which are important in practice) increased tenfold, but the run-time of one step of calculation, for example, at $N_V = 50\,000$ for a personal computer is more than 20 seconds, which is still not acceptable because problems which are practically interesting require to perform at least a few hundred or even thousand steps.

The only way to further computation acceleration is parallel computing technologies usage. MPI technology usage is preferred, because it is universal and allows to organize computation on shared-memory systems (in particular, on personal computers with multicore processors) as well as on cluster systems with distributed memory. Note that

when using OpenMP parallelization for shared memory systems with 2–8 cores the gain in computation time is not more than 2–5 % in comparison with MPI usage.

In Table 17 the rate of acceleration is shown for fast multipole method for $\theta = 0{,}2$ for some numerical schemes with different number of VEs.

N_V	Number of compute nodes					
	1	2	4	8	12	16
5 000	1,00	1,95	3,62	6,33	8,94	10,13
10 000	1,00	1,96	3,66	6,54	8,93	11,44
50 000	1,00	1,96	3,69	6,67	9,29	11,95
100 000	1,00	1,97	3,76	6,96	9,64	12,36

Table 17: Acceleration of computation when using MPI

For $N_V = 50\,000$ with 16 compute nodes run-time for one step of calculation is less than 2 seconds, and at $N_V = 100\,000$ — less than 4 seconds. It allows to use these algorithms in practice and solve actual problems. A further increase in the number of employed compute nodes allows to further reduce the computation time, because the algorithm is far from saturation as it can be seen from Table 17.

The calculation time for direct method without parallelization and fast multipole method on the 16-core cluster with acceptable accuracy of calculations ($\theta = 0{,}2$) is shown on Fig. 10. Computing time on both charts is referred to the time for problem with $N_V = 10\,000$, and the left graph on Fig. 10 is superimposed on graph of function $\bar{t} = 10^{-8} N_V^2$, and the right graph is superimposed on graph of function $\bar{t} = 1{,}2 \cdot 10^{-4} N_V$.

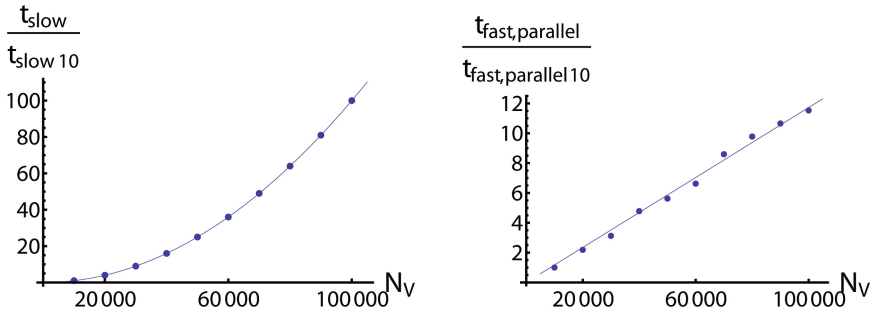

Figure 10: Computation time for different values of N_V when using direct method (left graph) and fast multipole method (right graph)

Linear increase of computation time is the result of two factors: fast multipole method provides the complexity of the algorithm proportional to $N_V \; lgN_V$, and the effectiveness of parallel algorithm with the growth of N_V increases slightly.

Graph of total acceleration achieved by using the fast multipole method and the parallel algorithm is shown on Fig. 11. Time of the problem solution using the direct method without parallelization was chosen as a basis time. It can be seen that the acceleration increases with N_V almost linearly and very close to superimposed graph of $u = 0{,}011 N_V$ function.

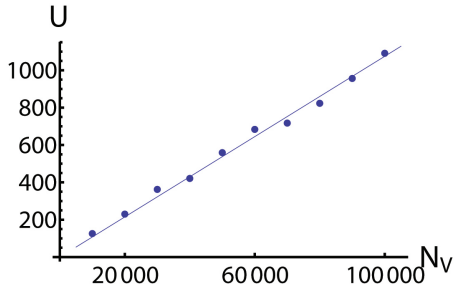

Figure 11: The achieved acceleration for different values of N_V compared with the calculation using direct method without parallelization

7 Conclusion

Methods of computation acceleration are investigated for spatial vortex structures evolution simulation using vortex element method. Model of symmetric vortex fragmenton allows to obtain sufficiently accurate results, nevertheless the numerical scheme should involve thousands and tens of thousands of vortex elements. It is shown that the known approaches related to parallel computing algorithms usage or fast multipole method usage (similar to one for gravitational N-body problem), if being applied alone don't provide computation acceleration, which would make it possible to solve such problems in practice. At the same time, the simultaneous application of both approaches makes it possible to achieve total acceleration in range $100 \ldots 1000$ times for the 16-core compute cluster. The execution time of one step of the calculation is about several seconds, and this allows to solve a large number of actual engineering problems in a reasonable time.

MPI parallel computing technology usage makes the developed algorithm and the program code based on it portable and versatile: computation can be performed both on multi-core systems with shared memory (efficiency reduce in comparison with OpenMP technology usage doesn't exceed a few percent) and on the cluster systems.

Acknowledgements

The author thanks Ph.D., D.Sc., Prof. G.A. Shcheglov for his support and attention to my research. The work was supported by Russian Federation Grant for young scientists [proj. MK-6482.2012.08] and Russian Fund of Basic Research Grant [proj. 11-08-00699-a].

References

[1] Barnes J., Hut P. A hierarchical $O(N \log N)$ force-calculation algorithm // Nature. 1986. V. 324. P. 446–449.

[2] Dynnikova G.Ya. Fast technique for solving the N-body problem in flow simulation by vortex methods // Computational Mathematics and Mathematical Physics. 2009. Vol. 49, N 8. P. 1389–1396.

[3] Lewis R.I. Vortex Element Methods For Fluid Dynamic Analysis Of Engineering Systems. Cambridge University Press, 2005.

[4] Cottet G.-H., Koumoutsakos P. Vortex methods: theory and practice. Cambridge University Press, 2000.

[5] Novikov E.A. Generalized Dynamics of Three-Dimensional Vortical Singularities (vortons) // Sov. Phys. JETP. 1983. Vol. 57. P. 566–569.

[6] Marchevsky I.K., Shcheglov G.A. 3D vortex structures dynamics simulation using vortex fragmentons // 6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), September 10-14, 2012, Vienna, Austria, Proceedings. Vienna University of Technology. 20 p.

[7] Alkemade A.J.Q. On vortex atoms and vortons. PhD Thesis, TU-Delft, April 1994.

[8] Bogomolov D.V., Marchevsky I.K., Setukha A.V., Shcheglov G.A. Numerical simulation of vortex rings pair motion in inviscid fluid using discrete vortex methods // Ingineering physics. 2008. Vol. 4. P. 8–14 [in Russian].

Ilia K. Marchevsky, Moscow, Russia