Proceedings of ALGORITMY 2016
pp. 93–102

ON THE EFFICIENCY OF FAST ALGORITHMS IN 2D VORTEX ELEMENT METHOD *

KSENIIA S. KUZMINA † AND ILIA K. MARCHEVSKY ‡

Abstract. The analogue of the Barnes — Hut algorithm is considered as one of the most efficient ways to acceleration of the velocities computation in vortex element method. When calculating the convective velocities this algorithm makes it possible to take into account the influence of vortex elements, which are located far enough from each other, approximately. The tree-based algorithm is developed for the calculation of diffusive velocities. The estimations of computational complexity are obtained for the algorithms for convective and diffusive velocities calculation. Also estimations for the errors of the vortex elements velocities computation are constructed, which depend on the algorithm parameters. These estimates allow in practice to choice the optimal parameters of the whole algorithm and achieve the maximum possible acceleration of the computations for the given maximum error level.

Key words. Vortex element method, Biot — Savart law, viscous flow, diffusive velocity, Barnes — Hut algorithm, binary tree, fast multipole method, computational complexity, error estimation

AMS subject classifications. 70F10, 76D17, 68Q25

1. Introduction. Vortex element method is meshfree Lagrangian CFD method which allows so simulate viscous and inviscid incompressible flows. It is very useful in simulation of the external flows when flow region is unbounded. The main advantages of vortex methods in such problems are small numerical dissipation and exact satisfaction of boundary conditions on infinity. For 'classical' mesh methods we should bound flow region and satisfy some artificial boundary conditions in order to simulate the external flow. Moreover, vortex method allows to simulate not only flows around immovable rigid bodies, but also hydroaeroelastic oscillations of bodies, and in this case its computational complexity remains nearly the same.

Vortex methods are very popular in various engineering applications because they allow to compute parameters of the flow as well as hydrodynamic (aerodynamic) forces acting the body. The accuracy of vortex methods usually is enough for practice, but their computational complexity is much lower in comparison with mesh methods, especially if approximate fast methods are used.

The aim of this paper is the accurate estimation constructing for computational complexity of fast algorithm of vortex methods and investigation of the dependence between the error and computational complexity.

2. Brief description of vortex methods. There are number of modifications of vortex methods for 2D and 3D flows simulations, sometimes they differ significantly one from another. Nevertheless, most of vortex methods have common trait. The primary flow variable is vorticity; its distribution in the flow region is simulated with large number of separate vortex elements — elementary vorticity fields. Each vortex element generates 'elementary' velocity field in whole flow region, and total velocity

*This work was supported by Grant No.: MK-3705.2014.8.

†Applied Mathematics Department, Bauman Moscow State Technical University, Russia, 105005, Moscow, 2-nd Baumanskaya st., 5 (kuz-ksen-serg@yandex.ru).

‡Applied Mathematics Department, Bauman Moscow State Technical University, Russia, 105005, Moscow, 2-nd Baumanskaya st., 5 (iliamarchevsky@mail.ru).

field is superposition of these 'elementary' fields. It is expressed by using the Biot — Savart law:

$$\vec{V}(\vec{r},t) = \vec{V}_\infty + \frac{1}{2(d-1)\pi}\int\limits_S \frac{\vec{\Omega}(\vec{\xi},t)\times(\vec{r}-\vec{\xi})}{|\vec{r}-\vec{\xi}|^d}dS_\xi, \tag{2.1}$$

where S is the part of the flow region where vorticity is non-zero, d is spacial dimension of the problem, $\vec{V}_\infty$ is the incoming flow velocity, $\vec{V}(\vec{r},t)$ is the flow velocity at the point with position vector $\vec{r}$ at the time moment t, $\vec{\Omega}(\vec{\xi},t) = \nabla\times\vec{V}(\vec{\xi},t)$ is vorticity vector in point with position vector $\vec{\xi}$.

The way of vorticity distribution approximation in the flow is determined by vortex element model choice. In 2D problems Rankine's circular vortices are normally used as vortex elements; in 3D case there are number of different models known, each of them has some shortcomings and advantages. Hereinafter we consider 2D case, however basic ideas could be also applied in 3D vortex methods after necessary adaptation.

In order to simulate 2D flow, we firstly write down the Navier — Stokes equation (or Euler equation) in Helmholtzs form by applying *curl* $(\nabla\times)$ operator to its left and right part:

$$\frac{\partial\vec{\Omega}}{\partial t} + (\vec{V}\cdot\nabla)\vec{\Omega} + \nu\Delta\vec{\Omega} = 0,$$

where ν is kinematic viscosity coefficient.

The easiest way to take into account the convective term $(\vec{V}\cdot\nabla)\vec{\Omega}$ is to move all the vortex elements along the streamlines, then their intensities (circulations) remain constant and the total velocity field at each time moment will correspond to flow of inviscid media.

In order to take into account viscous term, number of approaches are known: random walks method [1], particle strength exchange procedure [2], diffusive velocity method [3, 4, 5]. According to diffusive velocity approach, Navier — Stokes equations in 2D case can be written down in the following form

$$\frac{\partial\vec{\Omega}}{\partial t} = \nabla\times\big((\vec{V}+\vec{W})\times\vec{\Omega}\big), \quad \vec{\Omega} = \Omega\vec{k},$$

where $\vec{W} = -\nu\dfrac{\nabla\Omega}{\Omega}$ is the so-called diffusive velocity. So in order to simulate viscous flow according to Viscous Vortex Domains (VVD) method [4] we need to replace continuous vorticity distribution by N vortex elements with circulations Γ_i and solve numerically the following system at every time step:

$$\begin{cases} \dfrac{d\Gamma_i}{dt} = 0, \\ \dfrac{d\vec{r}_i}{dt} = \vec{V}(\vec{r}_i) + \vec{W}(\vec{r}_i), \end{cases} \qquad i = 1,\ldots,N.$$

The first equation means that intensities Γ_i of vortex elements remain constant (as in inviscid case), whereas their positions change as they move along the streamlines of summary $\vec{V}+\vec{W}$ velocity field.

3. Computational complexity of the vortex method. The main problem in vortex methods is convective velocity and diffusive velocity fields computation for all the vortex element. For Rankine's vortex element, according to the Biot — Savart law (2.1) for convective velocities we obtain

$$\vec{V}(\vec{r},t) = \vec{V}_\infty + \sum_{i=1}^{N} \frac{\Gamma_i}{2\pi} \frac{\vec{k} \times (\vec{r} - \vec{r}_i)}{\max\{|\vec{r} - \vec{r}_i|^2, \varepsilon^2\}}, \tag{3.1}$$

where ε is the radius of the vortex element, which assumed to be constant, $\vec{r}_i$ is position of i-th vortex element, Γ_i is its intensity, $\vec{k}$ is unit vector which is orthogonal to the plane of the flow. So if we estimate the computational complexity of the procedure of convective velocities computation for all the vortex elements, we easily find that it is proportional to N^2. To be more exact, it is $6N^2$ if we count only multiplication and division operations. In principle, it can be slightly reduced if we take into account that $\vec{V}_{ij}$ and $\vec{V}_{ji}$ have the opposite signs and differ only by Γ multipliers, but nevertheless its complexity is more than $3N^2$ ($\vec{V}_{ij}$ is the velocity from j-th vortex element at the point $\vec{r}_i$, where i-th vortex element is placed).

Direct (by using formula (3.1)) velocity computation is possible when number of vortex elements doesn't exceed some tens of thousands, otherwise time of computations becomes extremely high. Normally in order to simulate unsteady flow, especially in hydroelastic problems, we need to carry out thousands of time steps. If we want to raise the accuracy of flow simulation, we should raise number of vortex elements, which means that time step will become lower.

Note, that in this paper we don't touch the problem of flow simulation around airfoil, i.e. no-slip (no-throw in case of inviscid media) boundary condition satisfaction on its surface. In order to take it into account vorticity flux simulation is used in vortex method, according to which the airfoil is substituted by vortex (and source if the airfoil moves or it is deformable) layer which intensity can be determined from solution of some integral equation [6, 7, 8]. In general case the vortex layer becomes free, it also can be simulated by vortex elements, which become part of vortex wake.

In order to reduce time of simulation different fast techniques can be used. One of the most popular approaches is the Barnes — Hut [9] simulation usage which is similar to approximate fast algorithm for performing an N-body simulation. It had been adapted by G.Ya. Dynnikova for vortex method [10]. This algorithm is based on binary tree construction so that only vortex elements from nearby cells need to be treated individually, and vortices in distant cells can be treated as a single pair of vortices (with positive and negative intensities) centered at the cell's 'center of vorticity'. This can dramatically reduce the number of vortex pair interactions that must be computed. If number of levels of the tree is chosen correctly, then computational complexity of such an algorithm is proportional to $N \log N$. Number of vortex elements in the flow now can reach tens or even hundreds of thousands, and time of computation remains acceptable.

But in practice we need not only the order of computational complexity of the algorithm, but also some numerical estimations. As for the vortex methods, some early estimations [11, 12] allow to determine optimal number of tree levels, but their accuracy is not very high. In [13] authors derived much more accurate formula for

number of operations:

$$Q = \frac{24N^2}{2^k}\left(\frac{4}{\theta}\right)^2\left(1-\alpha\frac{(\sqrt{2})^k-1}{\sqrt{N}}\right)^2\left(1-\frac{4}{\theta(\sqrt{2})^k}\left(1-\alpha\frac{(\sqrt{2})^k-1}{\sqrt{N}}\right)\right)+ \\ +\frac{896\cdot 2^k\cdot\beta}{\theta^2}\left(4\left(\frac{1}{4+\theta}+\frac{1}{4-(\sqrt{2})^k\theta}\right)+\ln\left(\frac{(\sqrt{2})^k-4}{4+\theta}\right)\right)+4N. \tag{3.2}$$

In this estimation N is number of vortex elements in the flow, k is number of levels of the tree, θ is the accuracy parameter. Coefficients α and β are pure empirical, they depend on the problem being solved. The value of α ($0 < \alpha < 1$) mainly depends on the uniformity of vortex elements distribution in vortex wake, the value of β ($\beta > 0$) depends on the shape of vortex wake.

If value of θ is small ($0 < \theta \ll 1$), the accuracy of the fast algorithm is very high, but its computational complexity is very high too; the greater values of θ reduce the complexity significantly but the error becomes higher.

There are number of researches where the estimation for the error of Barnes — Hut method is obtained and proved mathematically [14, 15], but all these estimations have asymptotic behavior, they contain some undefined constants, so they are very useful for 'theoretical' estimations, but in practice they hardly can be used. The aim of this research is the derivation of the approximate estimations for computational complexity of the fast method and for the error of velocity computation.

4. Model problem. In order to obtain particular numerical results we consider the model problem of viscous flow simulation, which corresponds to the well-knows phenomenon of the viscous diffusion of circular vortex in unbounded region (Lamb's vortex). Let us assume that at initial time moment $t = 0$ there is vortex filament with circulation Γ (perpendicular to the flow plane) in the viscous incompressible media. The exact solution for vorticity distribution is

$$\Omega(r,t) = \frac{\Gamma}{4\pi\nu t}\exp\left(-\frac{r^2}{4\nu t}\right), \tag{4.1}$$

where r is distance to the center of the vortex. Total vorticity inside the circle of radius R is

$$\Gamma_R(t) = \int_0^{2\pi} d\varphi \int_0^R \Omega(r,\, t) r\, dr = \Gamma\left(1-\exp\left(-\frac{R^2}{4\nu t}\right)\right).$$

We consider that $\Gamma = 1$, kinematic viscosity $\nu = (2000\pi)^{-1} \approx 0.00016$, time moment at which we simulate vorticity distribution $t_0 = 2000\pi \approx 6283$. Then the circle of radius $R = 5$ contains more than 99.8 % of total vorticity, the other vorticity (outside this circle) can be neglected. The number of the vortex element we have used in order to simulate this vortex, was in interval $N = (60\,000,\, 90\,000,\, 120\,000, \ldots,\, 300\,000)$. Distribution of vortex elements was close to uniform. The example of such a distribution (with small number of vortex elements $N \approx 1000$) is shown on Fig. 4.1. Intensities of vortex elements were calculated analytically by integrating the exact solution (4.1).

In order to construct the tree for Barnes — Hut method 'standard' technique [10] was used, tree cells of different levels are shown on Fig. 4.2.

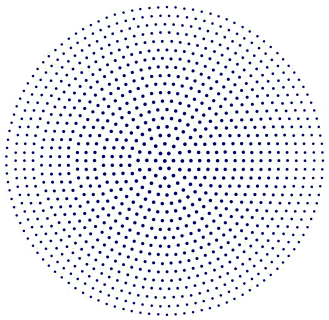

FIG. 4.1. *Distribution of vortex elements in circular vortex,* $N \approx 1000$

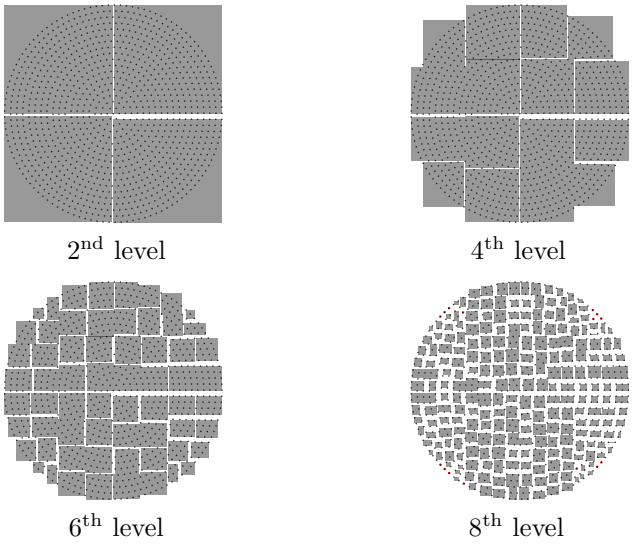

FIG. 4.2. *Cells from different tree levels*

5. Computational complexity estimation. In order to construct optimal numerical algorithm we need accurate estimation for the computational complexity of all its parts. Note, that tree construction is very 'light' procedure, even if number of vortex element has order of hundreds of thousand, in comparison with convective and diffusive velocities computation for all vortex elements.

5.1. Convective velocities computation. Computational complexity estimation of the algorithm of convective velocities calculation consists of two parts: first corresponds to direct velocity computation by using the Biot — Savart law for vortices which are close to the cell, which contains the observation point, and according to [13] it has the following form:

(5.1)
$$Q_{BS} = \frac{24N^2}{2^k}\left(\frac{4}{\theta}\right)^2\left(1-\alpha\frac{(\sqrt{2})^k-1}{\sqrt{N}}\right)^2\left(1-\frac{4}{\theta(\sqrt{2})^k}\left(1-\alpha\frac{(\sqrt{2})^k-1}{\sqrt{N}}\right)\right)+4N.$$

The other part is connected to approximate velocity computation from the cells,

which are far from observation cell:

$$Q_{Far} = \frac{896 \cdot 2^k \cdot \beta}{\theta^2} \left(4 \left(\frac{1}{4+\theta} + \frac{1}{4-(\sqrt{2})^k \theta} \right) + \ln \left(\frac{(\sqrt{2})^k - 4}{4+\theta} \right) \right). \tag{5.2}$$

The 'far distance' criterion has the following form:

$$\theta \cdot \delta > h + h_0, \tag{5.3}$$

here δ is the 1-norm ($\| \cdot \|_1$) of the vector between centers of the 'influence' and 'observation' cells, h and h_0 are sums of side lengths of these cells.

In order to obtain the estimation for α coefficient in (3.2) and (5.1), in test computations some special procedure had been introduced into the numerical algorithm, and number of operations was computed directly. The estimation for Q_{BS} (5.1) contains only α parameter and doesn't contain β, so it is possible to pick a good approximation for α.

The obtained results for number of operations Q_{BS} for vortex which contains 150 000 vortex elements in comparison with analytical estimation (5.1) for $\alpha = 0.8$ (for example) are shown on Fig. 5.1.

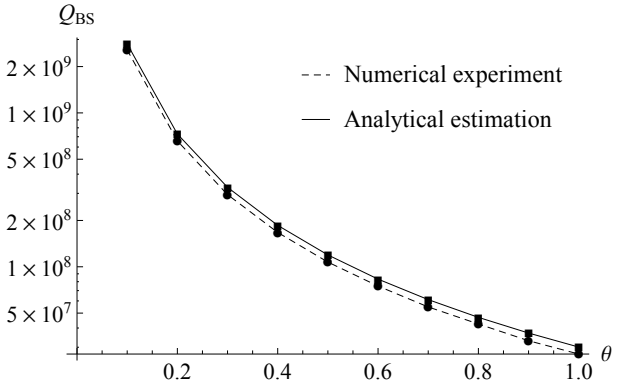

FIG. 5.1. *Number of operations Q_{BS} ($N = 150\,000$, $\alpha = 0.8$)*

In order to obtain optimal value of parameter α numerical experiments were carried out for different approximations of the initial vortex ($N = 60\,000 \ldots 300\,000$) and for ten values of θ parameter ($\theta = 0.1 \ldots 1.0$). Number of levels in the tree was chosen from the estimations [12, 13]. For every N coefficient α was chosen as the value, which minimizes sum of squares of the relative errors:

$$\sum_{i=1}^{10} \left(\frac{Q_{BS}(\theta_i, N, \alpha) - Q^*_{BS}(\theta_i, N)}{Q^*_{BS}(\theta_i, N)} \right)^2 \to \min,$$

where $\theta_i = 0.1\, i$, $Q^*_{BS}(\theta_i, N)$ is number of operations obtained from numerical experiment for N vortex elements and $\theta = \theta_i$.

The calculated optimal values of α are shown in Table 5.1.

The obtained values are very close one to the others, so in practice it is possible to use the average value $\alpha = 0.844$. In order to obtain the optimal value of parameter β in estimation (5.2), nearly the same procedure can be done for this estimation and

TABLE 5.1
Optimal values of α and β for different values of N

N	60 000	90 000	120 000	150 000	180 000	210 000	240 000	270 000	300 000
k	14	15	15	16	16	16	16	16	17
α	0.885	0.793	0.826	0.837	0.863	0.876	0.871	0.876	0.776
β	0.521	0.558	0.587	0.539	0.554	0.566	0.582	0.587	0.582

number of operations in numerical algorithm of convective velocity computation. The calculated optimal values of β are shown in Table 5.1. Again, these values are very close, so for practical purposes the average value $\beta = 0.561$ can be used.

So, the estimation for the computational complexity of the algorithm for convective velocities calculation is obtained.

5.2. Diffusive velocities computation. In order to compute diffusive velocities of vortex elements, we use nearly the same approach: we take into account influence only from the vortex elements which are placed in the tree cells of the last level, which satisfy the following condition:

$$\theta_{dif} \cdot \big(\delta - 0.5(h + h_0)\big) < \epsilon^*$$

Here θ_{dif} is 'far distance' criterion for diffusive velocities; the smaller it is the more accurate are the approximate results; δ, h, h_0 are the same as in (5.3); ϵ^* is typical distance between vortex elements. Note, that the necessary numerical formulae for diffusive velocity computation are derived, for example, in [4].

To derive the expression for estimation of the computational complexity of the algorithm for diffusive velocities calculation, we can use nearly the same reasonings as for convective velocities [13]. Omitting the intermediate computations (which sometimes are slightly cumbersome), we can obtain that the computational complexity is proportional to complexity Q of the algorithm for convective velocities (3.2), in which θ should by substituted by θ_{dif}:

$$Q_{dif} = Q\big|_{\theta=\theta_{dif}} \cdot N \cdot \theta_{dif} \cdot \frac{\gamma}{2^k}. \tag{5.4}$$

Here N is number of vortex elements, k is number of levels of the tree, γ is a coefficient which can be estimated numerically (Table 5.2).

TABLE 5.2
Optimal values of γ coefficient for different values of N

N	60 000	90 000	120 000	150 000	180 000	210 000	240 000	270 000	300 000
k	14	15	15	16	16	16	16	16	17
γ	0.0775	0.0775	0.0731	0.0712	0.0694	0.0689	0.0686	0.0679	0.0686

So, value of the coefficient γ is close to constant, and in order to obtain good approximation for computational complexity of the diffusive velocities calculation algorithm the average value $\gamma \approx 0.07$ can be used.

6. The errors of velocity calculation using fast algorithms. The main purpose of Barnes — Hut algorithm usage in vortex method is computations acceleration. The derived formulae (5.1), (5.2) and (5.4) make it possible to estimate how much time we need for flow simulation, but the main question is how much is the error of the fast method. Hereinafter for convective and diffusive velocities we compute

relative error

$$\epsilon = \frac{\max\limits_{i=1,\dots,N} \left|\vec{V}_i - \vec{V}_i^*\right|}{\left|\vec{V}_{conv}^*\right|},$$

where $\vec{V}_i$ is convective or diffusive velocity if the i^{th} vortex element calculated by using fast method; $\vec{V}_i^*$ is the same velocity computed directly (by using $O(N^2)$ algorithm), $|\vec{V}_{conv}^*|$ is maximum value of the convective velocities for all vortex elements. For the model problem which is considered in this paper, $|\vec{V}_{conv}^*| \approx 0.05$ and it almost doesn't depend on number of vortex elements N.

6.1. The error of convective velocity calculation. The calculation error of the convective velocity strongly depends on the value of the parameter θ; large values of the parameter θ correspond to large errors. The reasonable choice for parameter θ is value from interval $0 < \theta < 1$. In order to estimate its influence on the error of convective velocity computation, number of numerical experiments have been performed. Again, the calculations were carried out for $N = 60\,000 \dots 300\,000$ using the fast algorithm with different values of parameter $\theta = 0.1 \dots 1.0$. The obtained results are shown on Fig. 6.1. Thin lines correspond to the relative error for different N. The thick line is the majorizing curve.

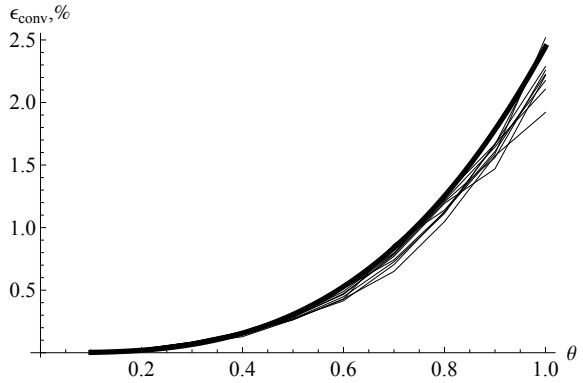

FIG. 6.1. *Relative error for convective velocities (thin lines correspond to $N = 60\,000 \dots 300\,000$, thick line — majorizing curve)*

It is easy to find that for each value of N the error dependence on θ can be approximated by function $\epsilon_{conv} = c \cdot \theta^3$ quite well, the value of coefficient c varies from 0.020 to 0.025. So the majorizing curve corresponds to

$$\epsilon_{conv} = 0.025 \cdot \theta^3$$

and this is sufficiently accurate estimation for the error of convective velocities computation using fast algorithm.

6.2. The error of diffusive velocity calculation. The error of diffusive velocity computation depends not only on the value of parameter θ_{dif}, but also on some other parameters. For each particular value of N it is easy to notice, that the relative error ϵ_{dif} is close to be quadratic function with respect to θ_{dif}. On Fig. 6.2 the corresponding curves are shown for $N = 90\,000$ and $N = 300\,000$. The value of the proportionality factor can be estimated by using least squares method.

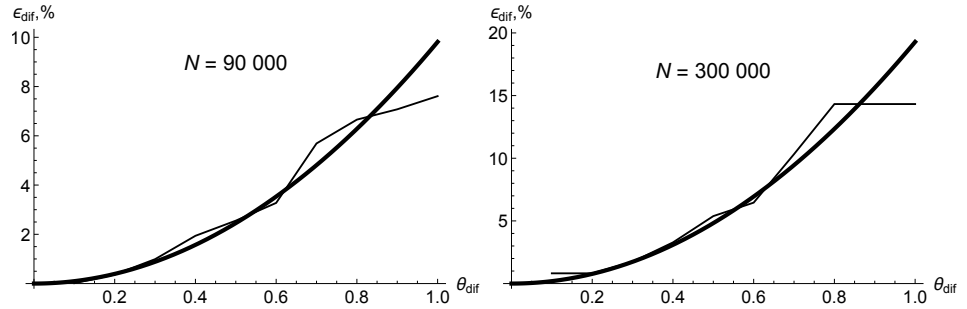

FIG. 6.2. *Relative error for diffusive velocities (thin lines — numerical experiment; thick line — quadratic function $\epsilon_{dif} = c_{dif}\theta_{dif}^2$)*

There is significant difference between the quadratic estimation and numerical experiment at high values of θ_{dif}, but for practical purposes it doesn't matter because the error of 10 or 20 % is unacceptable, so we should use much smaller values of θ_{dif} for which the correspondence with experiment is much better.

Calculating the values of c_{dif} factor for all test cases ($N = 60\,000 \ldots 300\,000$), we obtain the following results, shown in Table 6.1.

TABLE 6.1
Values of c_{dif} coefficient for different values of N

N	60 000	90 000	120 000	150 000	180 000	210 000	240 000	270 000	300 000
k	14	15	15	16	16	16	16	16	17
c_{dif}	6.86	9.81	9.76	13.70	13.76	13.19	13.51	13.72	19.28

It is clear from Table 6.1, that c_{dif} depends only on number of levels of the tree, i.e., $c_{dif} = c_{dif}(k)$, and for fixed k is almost constant. In principle, it is possible to write down some approximate formula for $c_{dif}(k)$ (by using least squares method or some nonlinear approximation), but it will be pure empirical and hardly usable in practice. The derivation of the analytical estimation for this factor is the aim of further research.

7. Conclusion. The accurate estimations for computational complexity of fast Barnes — Hut-type algorithm are derived for convective and diffusive velocities calculation procedures for vortex elements. For the model problem the error dependencies are obtained both for convective and diffusive velocities. If we assume, that the error level of $\epsilon = 0.2\,\%$ is acceptable, we can chose the following 'far criterions' in fast algorithm in vortex method:

$$\theta = 0.4, \qquad \theta_{dif} = 0.1.$$

Then we can estimate total computational complexity Q_{tot} of the fast algorithm for vortex method and compare it with $Q^*_{tot} = 15N^2$ from the 'direct' ($O(N^2)$) algorithm. The acceleration rate $\delta = Q^*_{tot}/Q_{tot}$ shows the efficiency of the fast method usage (Table 7.1).

So the acceleration rate for $N \approx 70\,000$ is about 100 times, and for $N \approx 270\,000$ is more than 350 times.

The derived estimations allow to choice the parameters of the numerical algorithm of vortex method in order to get the minimum possible computational complexity (i.e., time of computations) for the given level of the acceptable error.

TABLE 7.1
Computational complexities of the direct and fast algorithms and the acceleration rate

N	60 000	90 000	120 000	150 000	180 000	210 000	240 000	270 000	300 000
k	14	15	15	16	16	16	16	16	17
$Q_{tot}/10^9$	0.58	0.94	1.26	1.70	1.93	2.24	2.60	2.97	3.69
$Q^*_{tot}/10^9$	54.3	122.0	215.9	343.2	486.4	670.5	864.3	1097.4	1357.0
δ	93.6	129.8	171.3	201.9	252.0	299.3	332.4	369.5	367.8

REFERENCES

[1] A. J. CHORIN, *Numerical study of slightly viscous flow*, J. Fluid. Mech., 57 (1973), pp. 785–796.

[2] P. DEGOND, AND S. MAS-GALLIC, *The weighted particle method for convection-diffusion equations. Part I: The case of an isotropic viscosity*, Math. Comp., 53 (1989), pp. 485–507.

[3] Y. OGAMI, AND T. AKAMATSU, *Viscous flow simulation using the discrete vortex model-the diffusion velocity method*, Computers & Fluids, 19 (1991), pp. 433–441.

[4] G. YA. DYNNIKOVA, *Lagrange method for Navier — Stokes equations solving*, Doklady Akademii Nauk, 399 (2004), pp. 42–46.

[5] S. GUVERNYUK, AND G. DYNNIKOVA, *Modeling the flow past an oscillating airfoil by the method of viscous vortex domains*, Fluid Dynamics, 42 (2007), pp. 1–11.

[6] I. K. LIFANOV, AND S. M. BELOTSERKOVSKII, *Methods of Discrete Vortices*. CRC Press, 1993.

[7] S. N. KEMPKA, M. W. GLASS, J. S. PEERY, AND J. H. STRICKLAND, *Accuracy Considerations for Implementing Velocity Boundary Conditions in Vorticity Formulations*. SANDIA Report SAND96-0583, 1996.

[8] K. S. KUZMINA, AND I. K. MARCHEVSKY *The Modified Numerical Scheme for 2D Flow-Structure Interaction Simulation Using Meshless Vortex Element Method*, in Proc. IV Int. Conference on Particle-based Methods — Fundamentals and Applications (PARTICLES-2015), Barcelona (2015), pp. 680–691.

[9] J. BARNES, AND P. HUT, *A hierarchical $O(NlogN)$ force-calculation algorithm*, Nature, 324 (1986), pp. 446–449.

[10] G. YA. DYNNIKOVA, *Fast technique for solving the N-body problem in flow simulation by vortex methods*, Computational Mathematics and Mathematical Physics, 49 (2009), pp. 1389–1396.

[11] A. I. GIRCHA, *Fast Algorithm for N-body Problem Solving with Regard to Numerical Method of Viscous Vortex Domains*, Informatial technologies in Simulating and Control, 1 (2008), pp. 47–52 (in Russian).

[12] V. S. MOREVA, *On the Ways of Computations Acceleration when Solving 2D Aerodynamic Problems by Using Vortex Element Method*, Heralds of the Bauman Moscow State University. Natural Sciences. Sp.Issue 'Applied Mathematics' (2011), pp. 83–95 (in Russian).

[13] K. S. KUZMINA, I. K. MARCHEVSKY, *Estimation of computational complexity of the fast numerical algorithm for calculating vortex influence in the vortex element method*, Science & Education (electronic journal), 10 (2013), pp. 399–414 (URL: http://technomag.bmstu.ru/en/doc/604030.html)

[14] A. GRAMA, V. SARIN, AND A. SAMEH, *Improving Error Bounds for Multipole-Based Treecodes*, SIAM J. Sci. Comp., 21 (2000), pp. 1790–1803.

[15] J. K. SALMON, AND M. S. WARREN, *Skeletons from the treecode closet*, J. Comput. Phys., 111 (1994), pp. 136–155.