On estimates of computational complexity and error of the fast algorithm in the vortex methods

K S Kuzmina\textsuperscript{1,2}\textsuperscript{*} and V S Moreva\textsuperscript{2}

\textsuperscript{1}Bauman Moscow State Technical University, 105005, Russia, Moscow, \textsuperscript{2}Ivannikov Institute for System Programming of the RAS, 109004, Russia, Moscow,

\text{2}nd Baumanskaya str., 5

\text{2}nd Ivannikov Institute for System Programming of the RAS, 109004, Russia, Moscow, Alexander Solzhenitsyn st., 25

E-mail: kuz-ksen-serg@yandex.ru

Abstract. One of the efficient ways to speedup calculations in the vortex method, namely the Barnes – Hut-type algorithm, is considered. This method is based on the introducing of a hierarchical structure of domains (binary tree), which allows one to take into account approximately mutual influences of clusters of vortex elements located far from each other when calculating convective velocities. Estimates of the computational complexity of the algorithm for convective velocities calculating are derived, as well as estimates of the error, which depend on the parameters of the algorithm. In practice, these estimates make it possible to choose optimal values of the algorithm parameters and to achieve the maximal speedup of calculations at a given level of acceptable calculation error.

1. Introduction

Vortex method is a meshless Lagrangian method of computational fluid dynamics, which allows simulating the flows of an ideal or viscous incompressible medium. This method is very efficient in simulating of the external flow around airfoils when the flow domain is unbounded. The main advantages of vortex methods in solving such problems are low numerical dissipation and automatic satisfaction of the boundary condition at infinity, while when using mesh-based methods for simulating of the external flows, it is necessary to bound the computational domain and introduce some “artificial” boundary conditions. In addition, vortex methods make it possible to simulate not only flows around rigid bodies, but also simulate fluid-structure interaction (FSI), while maintaining approximately the same computational complexity.

Vortex methods are quite popular in various engineering applications. They make it possible to calculate both flow parameters and aerohydrodynamic forces acting on the body with sufficient for engineering purposes accuracy, while their computational complexity can be significantly lower compared to mesh methods, especially when using approximate fast algorithms in vortex methods. Such applications may include, inter alia, problems of industrial aerodynamics when it is required to evaluate wind loads acting on buildings and structures.

However, in order for their application to be truly effective, a high-quality software implementation of the algorithms used in vortex methods is required.

The aim of this work is to construct an estimate of the computational complexity of the fast algorithm for calculating the velocities of vortex elements in vortex methods and to the behaviour of the error of the approximate solution and the time of computations.
2. Brief description of vortex methods

There are several modifications of vortex methods for simulation two-dimensional and spatial flows, sometimes significantly different from each other both in terms of the implementation of particular operations and algorithms, and in the sense of the methods and approaches being used. However, their common feature is that the primary variable is vorticity, and its distribution in the flow domain is simulated by a large number of separate vortex elements. Each vortex element generates an “elementary” velocity field in the whole flow domain, and the general velocity field is a superposition of these “elementary” fields and is expressed according to the Biot – Savart law:

\[
\vec{V}(\vec{r},t) = \vec{V}_w + \frac{1}{2(\text{d}-1)\pi} \int_{S} \frac{\vec{\Omega}(\xi) \times (\vec{r} - \vec{\xi})}{|\vec{r} - \vec{\xi}|^2} \, dS_{\xi}
\]

(1)

Here \( S \) is the domain with nonzero vorticity; \( d \) is the spatial dimension in the considered problem; \( \vec{\Omega}(\xi, t) = \nabla \times \vec{V}(\xi, t) \) is the vorticity vector at the point with radius-vector \( \xi \).

Vorticity distribution in vortex element is determined by the choice of the vortex element model. In two-dimensional problems, the Rankin vortex model is usually used; for spatial problems, several models of vortex elements are known, each of which has its own advantages and disadvantages. We will consider two-dimensional case, but the basic ideas after some adaptation can be applied to solving spatial problems using vortex methods.

The viscous incompressible flow is described with the Navier — Stokes equations, which can be written down in the Helmholtz form:

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

where \( \nu \) is the kinematic viscosity coefficient.

The simplest way to take into account the convective term \((\vec{V} \cdot \nabla)\vec{\Omega}\) is to move all the vortex elements along the streamlines, while their intensities (circulations) remain constant; such an algorithm corresponds to the simulation of the ideal (inviscid) flow.

In order to take into account the term that corresponds to the viscosity influence, several approaches are known: the Random Walk method [1], the Particle Strength Exchange method [2], and the diffusive velocity method [3,4,5]. In this paper, the Viscous Vortex Domain (VVD) method based on the introduction of diffusive velocity is used [4,5]. According to this approach, the diffusive velocity field is introduced in the region with non-zero vorticity, and the Navier – Stokes equations in two-dimensional case can be written down in the following form:

\[
\frac{\partial \vec{\Omega}}{\partial t} = \nabla \times \left( (\vec{V} + \vec{W}) \times \vec{\Omega} \right), \quad \vec{\Omega} = \Omega \vec{k}
\]

Here \( \vec{W} = \nu \frac{\vec{\Omega}}{\Omega} \) is the diffusive velocity. So, when simulating a viscous flow by using the VVD method, it is necessary to solve numerically the following system at each time step:

\[
\begin{align*}
\frac{dI_i}{dt} &= 0, \\
\frac{d\vec{r}_i}{dt} &= \vec{V}(\vec{r}_i) + \vec{W}(\vec{r}_i),
\end{align*}
\]

\(i = 1, \ldots, N\)

The first equation means that the circulations of vortex elements \( I_i \) remain constant (as for case of ideal flow) while their positions change in such a way that they move along the streamlines of the summary velocity field \( \vec{V} + \vec{W} \).

3. Computational complexity of vortex method
The major part of computational complexity of vortex methods is connected with calculation of the convective and diffusive velocities for all the vortex elements. For Rankin vortices, according to the Biot – Savart law (1), we obtain the following expression for convective velocity:

$$\vec{V}(\vec{r},t) = \nabla \vec{\omega} + \sum_{i=1}^{N} \frac{\vec{k} \times (\vec{r}_i - \vec{r})}{2\pi \max|i \vec{r}_i|^2}$$

(2)

Here $\epsilon$ is a small radius of the vortex element that is assumed to be constant; $\vec{r}_i$ is the position of the vortex element; $l_i$ is its intensity; $N$ is number of vortex elements.

It is clear, that the computational complexity of the procedure of convective velocities calculation is proportional to $N^2$. If we consider only multiplication and division operations, it is equal to $6N^2$. It is possible to reduce complexity by taking into account that $\vec{V}_{ij}$ and $\vec{V}_{ji}$ have opposite signs and differ only by the factors $\Gamma$, however, the complexity is still exceed $3N^2$ ($\vec{V}_{ij}$ is the contribution of the $j$-th vortex element to the convective velocity of $i$-th vortex element).

Direct calculation of velocities using formula (2) is possible when the number of vortex elements does not exceed several tens of thousands, otherwise the calculation time becomes unacceptably large. Usually, to simulate an unsteady flow, especially in hydroelastic problems, it is necessary to perform up to several thousand time steps; the number of vortex elements can reach hundreds of thousands and even a million, which in turn will lead to an increase the calculation time of one time step.

This paper does not address the problem of modeling the flow around an airfoil, i.e. the no-slip boundary conditions at the airfoil surface line is not considered. We only note that to take into account the influence of the airfoil on the flow, the airfoil is replaced by a vortex sheet (and a source sheet in the case of a moving or deformable airfoil), the intensity of which can be determined from the solution of some integral equation [6,7,8].

To reduce the computational time, different methods can be used. One of the most popular approaches is the use of the Barnes – Hut-type algorithm [9], originally developed for an approximate fast solution of the gravitational $N$ body problem. This algorithm was adapted to vortex methods in [10]. It is based on the construction of a hierarchical structure of domains (tree). Directly (calculating the effect of each element on each), the vortex effect is calculated only from vortex elements that are located quite close to the cell under consideration, the effect of vortex elements located far from the considered cell is calculated approximately using simplified formulae.

Using this approach, one can achieve a significant reduction in the computational complexity of the algorithm: with the right choice of the number of tree levels, the complexity will be proportional to $N \log N$. This allows to solve problems with a large number of vortex elements in a reasonable time.

When solving practical problems, it is not enough to know only the order of computational complexity; a more detailed estimation is needed, which can help in choosing the optimal parameters of the algorithm. Known estimates [11,12] allow one to choose the optimal number of tree levels, but they have low accuracy. In [13], the authors obtained more accurate estimate for the number of operations:

$$Q = \frac{24N^2}{2^6} \left( \frac{\epsilon}{\vartheta} \right) \left[ 1 - \frac{4}{\theta \sqrt{2}} \left( 1 - \frac{\sqrt{\vartheta} - k - 1}{\sqrt{2}} \right) \right] \left[ 1 - \alpha \left( \frac{1}{\sqrt{2}} - k - 1 \right) \right]^{g + 1 - 1} + \frac{1}{4 + \frac{1}{\alpha \theta}} + \ln \left( \frac{1}{\alpha \theta} \right) + 4N(3)$$

Here $N$ is the number of vortex elements in the flow domain, $k$ is the number of tree levels, $\theta$ is the accuracy parameter. The values of the coefficients $\alpha$ and $\beta$ are determined empirically; they depend on the problem being solved. Value $\alpha$ ($\theta \ll \alpha < 1$) significantly depends on the distribution of vortex elements in the vortex wake, and $\beta$ ($\beta > 0$) depends on the shape of the vortex wake.

For small values of the parameter $\theta$ ($\theta \ll 1$), fast method has high accuracy, but in this case it also has high computational complexity; large values of $\theta$ can reduce the complexity of the calculations, but this increases the error of the solution.
There are works in which estimates and errors of the Barnes–Hut algorithm error were obtained and mathematically proved [14, 15], but these estimates are asymptotic and contain some unknown parameters. These estimates may be useful for a “theoretical” assessment, but in practice their application is hardly possible. The aim of this work is to construct approximate estimates for the computational complexity of the fast algorithm, as well as for the error of the velocities calculation.

4. Model problem

Let us consider the test problem of viscous flow simulation, which corresponds to the well-known phenomenon of diffusion of a circular vortex in an unbounded region (Lamb vortex). Let us suppose that at the initial time \( t = 0 \) an infinite vortex filament with circulation \( \Gamma \) is introduced into a viscous incompressible flow. The exact solution of this problem for the vorticity distribution at time \( t \) has the form:

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

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

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

We assume that \( \Gamma = 1 \), the kinematic viscosity coefficient of the flow \( \nu = (2000\pi)^{-1} \approx 0.00016 \), the start time of the simulation of the vorticity distribution is \( t_0 = 2000\pi \approx 6283 \). Then more than 99.8% of the total vorticity is contained in a circle of radius \( R = 5 \), the rest is neglected. The number of vortex elements that was used to simulate the distribution of vorticity inside this circle is taken from the interval \( N = 60000 ... 300000 \), their spatial distribution is close to uniform. An example of such a distribution (with a small number of vortex elements \( N \approx 1000 \)) is shown in Figure 1. The intensities of the vortex elements are calculated analytically by integrating the exact solution (4).

![Figure 1. Distribution of vortex elements in the circular vortex (N≈1000).](image)

To construct a tree for the Barnes–Hut type algorithm, the standard method was used [10], tree cells for different tree levels are shown in Figure 2.

![Figure 2. Cells for different levels of the tree.](image)
5. Estimation of the computational complexity

To build an efficient algorithm, a detailed assessment of the computational complexity of all its parts is necessary. Note that constructing a tree is a much easier procedure (even when the number of vortex elements of the order of hundreds of thousands) compared with the calculation of convective and diffusion velocities of all vortex elements.

The estimation of the computational complexity of the algorithm for calculating convective velocities consists of two parts: the first corresponds to the direct calculation of velocities using the Biot – Savart law from vortex elements whose cells are located close to each other, according to [13] this estimation has the following form:

\[
Q_{\text{BS}} = \frac{2N\pi^2}{2^4(\pi^2/\theta)} \left( 1 - \frac{\ln(\theta/\sqrt{2})}{\theta} \right) + 4N
\]

The second part of the estimate concerns the approximate calculation of velocities from cells that are sufficiently far from the considered cell:

\[
Q_{\text{Far}} = \frac{8\pi^2\theta^2}{\theta^2} \left( 4\left( \frac{1}{4+\theta} + \frac{1}{4\sqrt{2}^4} \right) + \ln \left( \frac{\sqrt{2}^4}{4+\theta} \right) \right)
\]

The criterion by which it is determined that the cells are placed far from each other has the following form:

\[
0.001\delta = h + h_0,
\]

where \(\delta\) is the octahedral norm \(||\cdot||_1\) of the vector connecting the centers of the “influencing” and “considered” cells, \(h\) and \(h_0\) are the sums of the lengths of the sides of these cells.

In order to obtain an estimate of the coefficient \(\alpha\) in formulae (3) and (5), an accurate calculation of the number of operations performed in the computation was performed. The estimate for \(Q_{\text{BS}}\) (5) contains only the parameter \(\alpha\), and the parameter \(\beta\) is not included in this expression, which allows us to obtain a good estimate for the parameter \(\alpha\).

The number of operations \(Q_{\text{BS}}\) obtained in the numerical experiment (the real number of operations) and using the analytical estimate (5) for 150 000 vortex elements with the parameter value \(\alpha=0.8\) are shown in Figure 3.

![Figure 3. Number of operations \(Q_{\text{BS}}\) (N=150000, \(\alpha=0.8\)).](image)

To obtain the optimal value of the parameter \(\alpha\), numerical experiments with different numbers of vortex elements (N = 60000…300000), simulating initial vortex, were carried out for different values of the parameter \(\theta = (0,1,0.1)\). The number of tree levels was chosen based on the estimates given in [12, 13]. For each value of \(N\), the coefficient \(\alpha\) was calculated that minimizes the sum of the squares of the relative errors:
\[
\sum_{i=1}^{10} \left( \frac{Q_{BS}(\theta_i,N) - Q_{BS}(\theta_i,N)^*}{Q_{BS}(\theta_i,N)} \right)^2 \rightarrow \min
\]

where \( \theta_i = 0.1 \); \( Q_{BS}(\theta_i,N) \) is the number of operations obtained from the numerical experiment for \( N \) vortex elements and \( \theta = \theta_i \).

The calculated optimal values of the parameter \( \alpha \) are given in Table 1. The obtained values are close to each other, so in practice, one can use the average value of \( \alpha = 0.844 \).

A similar procedure for calculating the number of operations and minimizing errors was performed to calculate the optimal value of the parameter \( \beta \) in the estimate (6). The obtained optimal values of the parameter \( \beta \) for different \( N \) are given in Table 1. Again, you can notice that these values are close, therefore, it is proposed, as in the case of \( \alpha \), to use the average value \( \beta = 0.561 \).

Table 1. Optimal values of \( \alpha \) and \( \beta \) for different \( N \) (\( k \) is the number of tree levels).

| \( N \) | 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 |
| \( \alpha \) | 0.885 | 0.793 | 0.826 | 0.837 | 0.863 | 0.876 | 0.871 | 0.876 | 0.776 |
| \( \beta \) | 0.521 | 0.558 | 0.587 | 0.539 | 0.554 | 0.566 | 0.582 | 0.582 | 0.582 |

So, an estimate of the computational complexity of the algorithm for calculating convective velocities is obtained.

6. Error velocity computation using the fast method

The main purpose of applying the Barnes-Hut algorithm is to speed up computations. The obtained formulae (5) and (6) allow us to estimate how much time is required to calculate convective velocities, but in addition, an important issue is the estimation of the error of the fast method. Next, we will consider the relative error in the calculation of convective velocities

\[
\varepsilon = \frac{\max_{i=1,...,N} \left| \mathbf{V}_i^\star - \mathbf{V}_i \right|}{\left| \mathbf{V}_{\text{conv}}^\star \right|}
\]

where \( \mathbf{V}_i \) is the convective velocity of the \( i \)-th vortex element calculated by the fast method; \( \mathbf{V}_i^\star \) is the velocity of the same vortex element, calculated directly by the formula (2); \( \mathbf{V}_{\text{conv}}^\star \) is the maximum value of convective velocities of all vortex elements. For the model problem considered in this paper, \( \mathbf{V}_{\text{conv}}^\star \), and this value is practically independent on the number of vortex elements \( N \).

The error of calculating the convective velocity significantly depends on the value of the parameter \( \theta \); Larger values of the parameter \( \theta \) correspond to larger errors. In order to evaluate how the parameter \( \theta \) affects the error in calculating the convective velocity, a number of computational experiments were performed. Calculations were performed for \( N = 60000 \ldots 300000 \) using the fast method with different parameters \( \theta = 0.1 \ldots 1.0 \). The results are shown in Figure 4; thin lines correspond to the relative errors in the calculation of convective velocity for experiments with different \( N \), the thick line corresponds to the curve, which is their upper estimate.
It can be noted that for each value of $N$, the dependence of the error on $\theta$ is well approximated by the function $e_{\text{conv}} = c \cdot \theta^3$, and for different $N$ the value of the coefficient $c$ varies from 0.020 to 0.025. Thus, the majorant of these curves

$$e_{\text{conv}} = 0.025 \cdot \theta^3$$

corresponds to a fairly accurate estimate of the error in calculating convective velocities by the fast method.

7. Conclusions
A detailed estimation of the computational complexity of an algorithm of the Barnes – Hut type for the procedure for calculating convective velocities in the vortex method is described. The dependence of the error in calculating convective velocities for the model problem is obtained. If we accept the permissible error $\varepsilon = 0.2\%$, then we can choose the following values of the range parameters for the fast method algorithm: $\theta = 0.4$. We calculate the total computational complexity $Q$ of calculating convective velocities by the fast method and compare it with the complexity of the direct method $Q^* = 6N^2$ operations. The speedup value $\delta = \frac{Q}{Q^*}$ shows the efficiency of using the fast method (Table 2).

| $N$   | 60 000 | 90 000 | 120 000 | 150 000 | 180 000 | 210 000 | 240 000 | 270 000 | 300 000 |
|-------|--------|--------|---------|---------|---------|---------|---------|---------|---------|
| $k$   |        |        |         |         |         |         |         |         |         |
| $10^{-9}$ | 0.35  | 0.62  | 0.82   | 1.13   | 1.27   | 1.43   | 1.67   | 1.91   | 2.64   |
| $Q^*10^{-9}$ | 21.72 | 48.8  | 86.4   | 137.3  | 194.6  | 268.2  | 345.7  | 439.0  | 542.8  |
| $\delta$ | 62.6  | 78.2  | 105.0  | 120.8  | 153.2  | 186.5  | 207.3  | 230.0  | 205.3  |

The estimates obtained make it possible to choose parameters for the fast algorithm to obtain the minimum possible computational complexity for the required error.

Note, that the developed approach can be generalized for spatial (3D) flows simulation using vortex methods. Its Preliminary estimates shows, that in 3D problems it would be much more efficient in comparison to 2D case, since number of vortex elements (i.e., vortex lines, which form closed vortex loops in method, described in [16]) even for the coarse modelling exceeds tens of thousand.

Acknowledgment
This research is supported by Russian Foundation for Basic Research (project 18-31-00245).

References
[1] Chorin A J 1973 *J. Fluid Mech.* **57** 785 doi: 10.1017/S0022112073002016
[2] Degond P and Mas-Gallic S 1989 *Math. Comp.* **53** 485 doi: 10.1090/S0025-5718-1989-0983559-9
[3] Ogami Y and Akamatsu T 1991 *Computers and Fluids* **19** 3/4 433 doi: 10.1016/0045-7930(91)90068-S
[4] Dynnikova G Ya 2004 *Doklady Physics* **49** 11 648
[5] Guvernyuk S and Dynnikova G 2007 *Fluid Dynamics* **42** 1 doi: 10.1134/S0015462807010012
[6] Lifanov I K and Belotserkovskii S M 1993 *Methods of Discrete Vortices* (CRC Press)
[7] Kempka S N, Glass M W, Peery J S, Strickland J H and Ingber M S 1996 *SANDIA Report* SAND96-0583 doi: 10.2172/242701
[8] Kuzmina K S and Marchevsky I K 2019 *Fluid Dynamics* **54** 7 991 doi: 10.1134/S0015462807010012
[9] Barnes J and Hut P 1986 *Nature* **324** 446 doi: 10.1038/324446a0
[10] Dynnikova G Y 2009 *Comp. Math. and Math. Physic* **49** 8 1389 doi: 10.1134/S0965542509080090
[11] Gircha A I 2008 *Informatsionniye tekhnologii modelirovaniya i upravlemiya* 1 47 (in Russian)
[12] Moreva V S 2011 *Herald of the Bauman Moscow State Technical University. Series Natural Sciences. Special Issue “Applied mathematics”* 83 (in Russian)
[13] Kuzmina K S and Marchevsky I K 2013 Science and Education **10** 399 (in Russian) doi: 10.7463/1013.0604030
[14] Grama A, Sarin V and Sameh A 2000 *J. Sci. Comp.* **21** 1790
[15] Salmon J K and Warren M S 1994 *J. Comput. Phys.* **111** 136
[16] Dergachev S A, Marchevsky I K and Shcheglov G A 2019 *Aerospace Science and Technology* **94** 105374 doi: 10.1016/j.ast.2019.105374