BARNES–HUT/MULTIPOLE FAST ALGORITHM IN THE LAGRANGIAN VORTEX METHOD

ALEXANDRA KOLGANOVA, ILIA MARCHEVSKY & EVGENIYA RYATINA
Bauman Moscow State Technical University, Russia

ABSTRACT

New modification of the fast algorithm based on the Barnes–Hut (BH) and fast multipole method (FMM) is developed for the problem of velocities calculation in vortex particle method. It provides a quasilinear computational complexity and allows for the accuracy flexible adjustment, similar to the classic Barnes–Hut method. Four schemes are developed with a different number of terms being held in multipole and local expansions. All the necessary formulae are presented, expressed in terms of operations with complex numbers. If extremely high accuracy is not required, the proposed algorithm is more efficient in comparison to the traditional FMM methods.

Keywords: fast algorithm, n-body problem, vortex method, the Barnes–Hut algorithm, multipoles.

1 INTRODUCTION

Vortex particle method [1], [2] is an efficient tool for solving problems of incompressible flow simulation around airfoils including fluid structure interaction problems. The main idea is to consider the vorticity $\vec{\Omega} = \mathrm{curl}\vec{V}$ as a primary computational variable. For the viscosity effect simulation, we use the viscous vortex domains (VVD) method, based on the diffusive velocity approach, that seems to be the most efficient [3]. Thus, the Navier–Stokes equations in 2D case for incompressible flow can be written down in vorticity form:

$$\frac{\partial\vec{\Omega}}{\partial t}+\nabla\times(\vec{\Omega}\times(\vec{V}+\vec{W}))=\vec{0}, \tag{1}$$

where $\vec{V}$ is flow velocity, $\nabla\cdot\vec{V}=0$; $\vec{\Omega}=\Omega\vec{k}$; $\vec{k}$ is unit vector orthogonal to the flow plane; $\vec{W}=-\nu\frac{\nabla\Omega}{\Omega}$ is the so-called diffusive velocity, proportional to the kinematic viscosity coefficient ν.

Eqn (1) can be considered as the transfer equation for vorticity that moves with the velocity $(\vec{V}+\vec{W})$. Vorticity field in the flow domain is simulated by a set of N vortex particles – Rankine or Lamb vortices, characterized by positions in the flow domain $\vec{r}_j$, circulations Γ_j that remain constant in time, $j=1,\ldots,N$, and small radius ε, equal for all the vortices.

The velocity field can be reconstructed using the Biot–Savart law. If there are no airfoils in the flow domain, it takes the form

$$\vec{V}(\vec{r})=\vec{V}_\infty+\int_S\frac{\vec{k}\times(\vec{r}-\vec{\xi})}{2\pi|\vec{r}-\vec{\xi}|^2}\Omega(\vec{\xi})dS_\xi=\vec{V}_\infty+\sum_{j=1}^{N}\frac{\Gamma_j}{2\pi}\frac{\vec{k}\times(\vec{r}-\vec{r}_j)}{\max\{|\vec{r}-\vec{r}_j|^2,\varepsilon^2\}}, \tag{2}$$

here $\vec{V}_\infty$ is the incident flow velocity; Rankine vortex model is considered.

WIT Transactions on Engineering Sciences, Vol 135, © 2023 WIT Press
www.witpress.com, ISSN 1743-3533 (on-line)
doi:10.2495/BE460131

The vortex convective velocities calculation $\vec{V}(\vec{r}_i)$, $i=1,\ldots,N$, according to eqn (2) is the most time-consuming operation in the vortex particle method algorithm. Being performed directly, its computational complexity is proportional to N^2; this problem is similar to the gravitational N-body problem. Taking into account that in vortex method the number of particles can reach millions [4], the only efficient way to reduce computational complexity is approximate fast methods implementation, which have quasilinear complexity. In order to choose optimal fast method, it is necessary to take into account specific features of the problem, the required accuracy, consistency of particular fast method with implementation of other operations in the whole algorithm, its scalability (for parallel computers of different architectures), etc. Note, that the considered problem differs significantly from the classical N-body problem: firstly, the kernel function in the integrand in eqn (2) decreases proportionally to distance, instead of squared distance, and secondly, vortex particles circulations can be positive or negative while body masses are always positive.

The tree-based Barnes–Hut method is the first fast method, suggested in 1986 for such problems [5], and it still remains popular. It has quasilinear computational complexity $O(N\log N)$ instead of squared one $O(N^2)$. Its adaptation for velocities calculation in vortex particle method is suggested in Dynnikova [6]. However, there are several issues: it is almost always less efficient in comparison to the fast multipole method (FMM) [8]; its adaptation for other operations in vortex method algorithm, e.g., the boundary integral equation solution, is not very efficient. Note also, that it can not be generalized for 3D flows simulation. The FMM algorithm, suggested slightly later in 1987, is also tree-based, but the tree traversal is performed in quite different way in comparison to the previously mentioned Barnes–Hut method. It has linear $O(N)$ computational complexity. Note, that it is very efficient when the extremely large number of particles is considered or very high, up to machine precision is required. But it is not obvious, that it is preferable in all the cases.

For example, in Capuzzo-Dolcetta and Miocchi [10], the authors had compared two methods for the gravitational problem by varying the accuracy and number of particles, and stated that the Barnes–Hut algorithm is more efficient if N has an order of hundreds of thousands. However, the multipole approximation works good for the clusters of dense-placed particles. If the problem implies essential non-uniform distribution of particles, hybrid mesh-hierarchical methods are known [11] that combine fast mesh algorithms (known as P3M [12]) for areas with rarefied-placed particles and multipole approximation for clusters. The ideas of constructing hybrid FMM/BH algorithm are also known [13] where high-order expansion of the influence function is used for electrostatic potential calculation, but without its local expansions. Taking into account that in vortex particle method the extremely high accuracy is not required, and the number of particles in 2D problems can reach a few million, the combination of the FMM and BH methods can lead to the optimal solution for the considered problems. This idea can be considered as the development of one proposed in Dynnikova [6] from more general point of view.

In the present paper a new modification of the Barnes–Hut algorithm is proposed, that has hybrid nature and includes some ideas of the Fast Multipole Method. The suggested algorithm seems to be preferable for 2D flows simulation due to flexible accuracy adjusting and convenient tree structure.

2 THE BARNES–HUT/MULTIPOLE ALGORITHM

The suggested method belongs to the tree-codes. The tree is the hierarchical structure of rectangular cells. The root cell includes all the particles in the flow domain. Then it is divided across the longest side, and two resulting first-level cells are cropped precisely to the

particles, that are contained in them. In contrast to the original Barnes–Hut method, tree cells dividing is performed not up to level at which each cell contains a single vortex, but the process is stopped on the specified tree level. The example of tree cells for the vortex wake after the bridge cross section is shown in Fig. 1. The vortex wake is simulated with vortex particles.

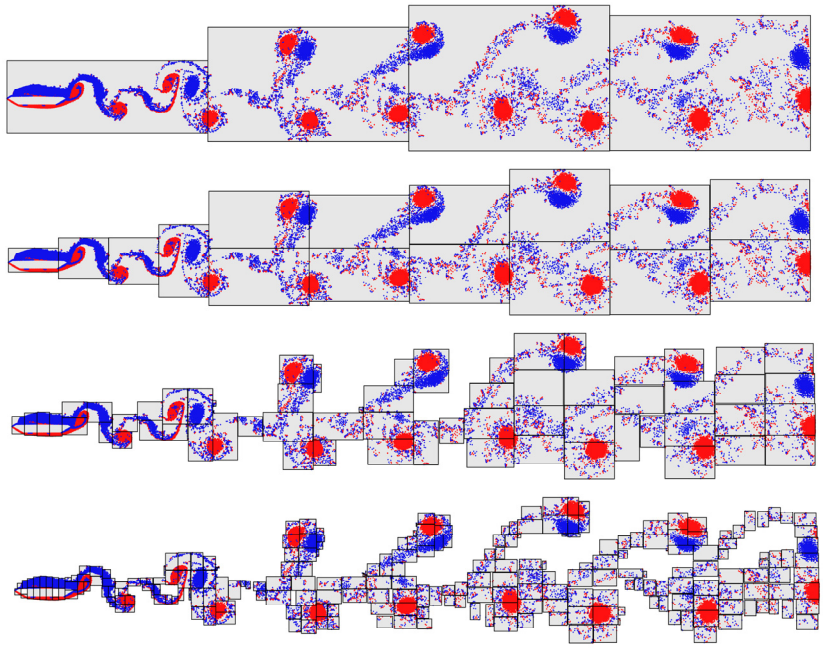

Figure 1: Tree cells example: 2nd, 4th, 6th and 8th level cells.

Note, that the above described algorithm of tree constructing can be called 'naive' and has the only advantage: it is adaptive, that is clearly seen in Fig. 1. However, it is recursive and poorly scalable; in parallel mode three construction for large values of N becomes very time-consuming procedure, takes up to half of time of the entire algorithm. Much more efficient way is to use non-recursive algorithm based on the fractal Morton's curve construction [7].

The main idea of the method suggested in the present paper, is that the influence of some far-placed cluster of particles onto another cluster, called 'control cluster' (more precisely, onto its centre), is calculated approximately using multipole expansion of the influence function. Then, the local expansion of the influence function is constructed in the control cluster, that is finally used for velocities computation at vortex particles positions. The velocity, induced by the vortex particles, placed in closely-placed clusters, is calculated directly according to the Biot–Savart law (2).

The method has two adjustable parameters: tree depth, that should be chosen in such a way to minimize computational complexity of the whole algorithm, and proximity parameter θ, that determines the ratio between the accuracy and computational complexity. We suppose that the cell p is far enough from another cell q if the distance between their centers $|\vec{\rho}|$ satisfies the proximity criterion:

$$|\vec{\rho}| \geq \frac{h_p + h_q + \varepsilon}{\theta}, \tag{3}$$

where h_p and h_q are sums of width and length of the corresponding cells; vortices radius ε is added to numerator for the correct processing of cells containing a single particle, since the value of h for such cells is equal to zero.

2.1 Main idea of the modification

Let us consider firstly the model problem of the velocities calculation at the points, at which the particles are placed in a control cluster, that are influenced by vortex particles placed in an influence particle (Fig. 2). We assume, that the control and influence clusters are far enough, so that the criterion (3) is satisfied. However, the resulting formulae and the numerical algorithm will be written down for the general case.

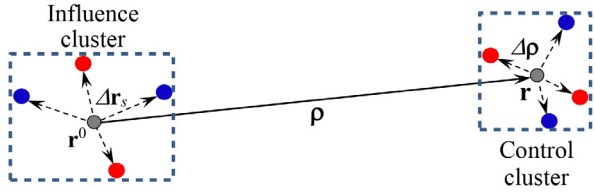

Figure 2: Scheme of two far-placed clusters interaction.

Here the following notations are introduced: $\vec{r}^{\,0}$ and $\vec{r}$ is the influence and control clusters centre positions, respectively; $\vec{\rho} = \vec{r} - \vec{r}^{\,0}$; $\vec{r}_s$ is the sth particle position in the influence cluster ($s = 1,\ldots,N_s$); $\Delta\vec{r}_s = \vec{r}_s - \vec{r}^{\,0}$; $\Delta\vec{\rho}$ is the vortex particle position in the control cluster with respect to its center.

The direct influence (hereinafter the 'influence' means the velocity, induced by the considered influence cluster at the corresponding point) at the control cluster centre can be calculated according to the Biot–Savart formula:

$$\vec{V}(\vec{r}) = \sum_{s=1}^{N_s} \Gamma_s \vec{k} \times (-\nabla_r G(\vec{r} - \vec{r}_s)), \qquad G(\vec{r}) = \frac{1}{2\pi} \ln \frac{|\vec{d}_*|}{|\vec{r}|},$$

here $\vec{d}_*$ is arbitrary constant non-zero vector.

Assuming that $|\Delta\vec{r}_s|$ is much smaller than $|\vec{\rho}|$, the influence function $\vec{V}(\vec{r})$ can be expanded into Taylor series with respect to the variables $|\Delta\vec{r}_s|/|\vec{\rho}|$:

$$\vec{V}(\vec{r}) = \sum_{s=1}^{N_s} \Gamma_s \vec{k} \times (-\nabla_r G((\vec{r} - \vec{r}^{\,0}) - \Delta\vec{r}_s)) = -\sum_{s=1}^{N_s} \Gamma_s \vec{k} \times (\nabla_\rho G(\vec{\rho} - \Delta\vec{r}_s))$$
$$\approx \vec{V}^m(\vec{\rho}) + \vec{V}^d(\vec{\rho}) + \vec{V}^q(\vec{\rho}) + \vec{V}^o(\vec{\rho}) + \vec{V}^h(\vec{\rho}), \tag{4}$$

where $\vec{V}^m$, $\vec{V}^d$, $\vec{V}^q$, $\vec{V}^o$ and $\vec{V}^h$ are monopole, dipole, quadrupole, octupole and hexadecapole terms respectively, calculated as

$$\vec{V}^{(p)}(\vec{\rho}) = \frac{(-1)^{p+1}}{p!}\sum_{s=1}^{N_s}\Gamma_s\vec{k}\times(\nabla_\rho(\underbrace{\nabla_\rho\ldots\nabla_\rho}_{p\ times}G(\vec{\rho})\cdot\ldots\cdot\underbrace{\Delta\vec{r}_s\otimes\ldots\otimes\Delta\vec{r}_s}_{p\ times})), \tag{5}$$

where the operation '$\cdot$' means inner tensor product; '$\otimes$' is outer (Kronecker) tensor product; the superscript '(p)' corresponds to the p-th multipole term: $p=0$ for the monopole term, $p=1$ for the dipole term, etc. In fact, in the original Barnes — Hut algorithm monopole and dipole terms are taken into account because the centre of expansion is chosen at the centre of mass (the dipole moment in this case is equal to zero). Since vortex particles can have both positive and negative circulations, the vorticity centre, being understood similarly to the center of mass, can lie outside the cell or even at infinity (the last case takes place when total vorticity in the cell is equal to zero). Moreover, the accuracy of the expansion depends on position of expansion center. Thus, we choose the center of expansion at the geometrical cell center, that provides minimal a'priori error estimation. Another possible way is to consider vortices with positive and negative circulations separately, as it is suggested in Dynnikova [6]; in this case it is necessary to perform all the operations twice.

After series expansion of the influence function (4), all the multipole terms (5) should be factorized in order to distinguish the corresponding multipole moments that depend only on the s th cell parameters, hence they are calculated once for each cell. The general factorized formula is the following:

$$\vec{V}(\vec{r}) \approx \sum_p \vec{V}^{(p)}(\vec{\rho}) = \sum_p(\overline{\vec{\Theta}}^{(p)}(\vec{\rho})\cdot\ldots\cdot\overline{\vec{m}}^{(p)}(\Delta\vec{r}_s,\Gamma_s)),$$

where the multipole moments $\overline{\vec{m}}^{(p)}$ are the p th rank tensors (scalar value $m^{(0)}$ is considered as the 0th rank tensor); the coefficients $\overline{\vec{\Theta}}^{(p)}$ are the $(p+1)$ th rank tensors equal to logarithmic potential multiple gradients up to constant factor $\pm 1/(2\pi)$:

$$\overline{\vec{\Theta}}^{(p)}(\vec{\rho}) = (-1)^{p+1}\underbrace{\nabla_\rho\ldots\nabla_\rho}_{(p+1)\ times}(\ln\frac{|\vec{d}_*|}{|\vec{\rho}|}), \quad p=0,\ldots,4. \tag{6}$$

Here the notation $\overline{(\cdot)}$ is used for a tensor of arbitrary rank.

In the framework of traditional Barnes–Hut algorithm, a single particle is contained in each leaf cell, whereas the consideration of multiple particles per leaf cell requires much smaller tree (in term of tree depth), and therefore can be more efficient [9]. The optimal tree depth, that provides minimal computational complexity, depends on number of vortex particles as well as on geometrical shape of the vortex wake. Note, that the tree depth almost does not effect the accuracy of the velocities computation.

Thus, for vortex particles velocities calculation in the suggested fast method modification, the local expansions of the multipole terms in eqn (4) should be performed with respect to small distance $|\Delta\vec{\rho}|$ (Fig. 2). As a result, the velocity is calculated as follows:

$$\vec{V}(\vec{r}+\Delta\vec{\rho}) \approx \sum_p((\overline{\vec{\Theta}}^{(p)}(\vec{\rho}) + \nabla_\rho\overline{\vec{\Theta}}^{(p)}(\vec{\rho})\cdot\Delta\vec{\rho}+\ldots)\cdot\ldots\cdot\overline{\vec{m}}^{(p)}(\Delta\vec{r}_s,\Gamma_s)).$$

This expression can be transformed and written down in suitable for practical computations form by grouping the terms containing multipliers with the same $\Delta\vec{\rho}$ degree (we denote $\Delta\vec{\rho}^{\otimes 2} = \Delta\vec{\rho} \otimes \Delta\vec{\rho}$, etc.):

$$\vec{V}(\vec{r} + \Delta\vec{\rho}) \approx \frac{\vec{k}}{2\pi} \times (\vec{E}_0(\vec{\rho}) + \hat{\vec{E}}_1(\vec{\rho}) \cdot \Delta\vec{\rho} + \frac{1}{2}\underline{\vec{E}}_2(\vec{\rho}) \cdot\cdot \Delta\vec{\rho}^{\otimes 2} + \frac{1}{6}\vec{E}_3(\vec{\rho}) \cdots \Delta\vec{\rho}^{\otimes 3}), \tag{7}$$

where coefficients $\overline{\vec{\vec{E}}}_q$ are tensors of the $(q+1)$ th rank, which are accumulated for each control cluster over all influence clusters that are far enough from it according to the criterion (3). As it has been mentioned, the tensors $\overline{\vec{\vec{E}}}_q$ are obtained as the coefficients in power (local) expansions of multipole terms, where the vector $\vec{E}_0$ corresponds to the sum of constant terms, the matrix $\hat{\vec{E}}$ – to linear terms, etc. Note, that number of terms in local expansion of multipole terms in (4) should be chosen consistently.

For example, if one considers only monopole term in eqn (4), there is no need to perform its local expansion at all: linear term of the local expansion has the same order of magnitude as the omitted dipole term. Then, if monopole and dipole terms are taken into account, the first one should be expanded up to linear term, while the second one can be considered as a constant, etc. However, in the table shown below, it is suggested to consider one additional multipole term (as constant, without local expansion). This approach allows to deal with influence clusters, which are much bigger than the control one; such situation is very common at the tree traversal. So, the following table shows the coefficients k_i, which are equal to 1 if the corresponding coefficient should be taken into account; otherwise they are equal to 0. The first row marked by I, corresponds to the coarsest approximation, where only two terms are taken into account without their local expansions; row II corresponds to more accurate version, etc.

	k_1	k_2	k_3	k_4
I	1	0	0	0
II	1	1	0	0
III	1	1	1	0
IV	1	1	1	1

For the most accurate scheme IV the tensor coefficients $\overline{\vec{\vec{E}}}_q$ in local expansion are calculated using the following formulae:

$$\vec{E}_0(\vec{\rho}) = \vec{\vec{\Theta}}^m m^m + k_1\hat{\vec{\Theta}}^d \cdot \vec{m}^d + \frac{k_2}{2!}\underline{\vec{\Theta}}^q \cdot\cdot \hat{\vec{m}}^q + \frac{k_3}{3!}\vec{\Theta}^o \cdots \underline{\vec{m}}^o + \frac{k_4}{4!}\tilde{\vec{\Theta}}^h \cdots \tilde{\vec{m}}^h,$$

$$\hat{\vec{E}}_1(\vec{\rho}) = -k_2\hat{\vec{\Theta}}^d m^m - k_3\underline{\vec{\Theta}}^q \cdot \vec{m}^d - \frac{k_4}{2!}\vec{\Theta}^o \cdot\cdot \hat{\vec{m}}^q, \tag{8}$$

$$\underline{\vec{E}}_2(\vec{\rho}) = k_3\underline{\vec{\Theta}}^q m^m + k_4\vec{\Theta}^o \cdot \vec{m}^d,$$

$$\vec{E}_3(\vec{\rho}) = -k_4\vec{\Theta}^o m^m.$$

Eqn (7) is the final formula for approximate calculation of vortex particles velocities in control cluster, induced by far-placed influence clusters. However, calculations according to the presented formulae are extremely time-consuming due to operation on a lot of components of higher rank tensors. At the same time, all the tensors $\overline{\vec{m}}^{(p)}$, $\overline{\vec{\Theta}}^{(p)}$ and $\overline{\vec{E}}_q$ are fully symmetric (with respect to any pair of indices) and their convolutions over arbitrary pair of indices are equal to zero, so all of them (except monopole moment m^m which is scalar) are determined by only a pair of real numbers. Thus, in practice it is necessary to store and calculate only two components of each tensor.

2.2 Formulae for numerical calculations

Since we deal with 2D flow simulation, the vortex particle position in the flow domain $\vec{r}_s = \{x_s, y_s\}$ can be considered as a complex number $z_s = x_s + iy_s$, where i is the unit imaginary number. In further formulae we use the operations of complex numbers multiplication $z_1 \cdot z_2$ and raising to a power z^p.

Let us firstly consider the multipole moments calculation of tree leaf cells. The monopole moment is real scalar itself, so we still use the notation m^m. The tensor of any higher multipole moment $\overline{\vec{m}}^{(p)}$ is determined by two real numbers, denoted as $m_0^{(p)}$ and $m_1^{(p)}$, which can be considered as a real and imaginary parts of the complex number $m^{(p)} = m_0^{(p)} + im_1^{(p)}$. Thus, the multipole moments can be represented through vortex circulations Γ_s and complex numbers z_s that correspond to their positions $\Delta\vec{r}_s$, $s = 1,\ldots,N_s$, as follows:

$$m^{(p)} = \sum_{s=1}^{N_s} \Gamma_s z_s^p, \quad p \in \mathbb{N}. \tag{9}$$

Then the parent-cell parameters can be found by summation of children multipole moments (represented through complex numbers), being shifted to the centre of parent cell. The shifting scheme is shown in Fig. 3.

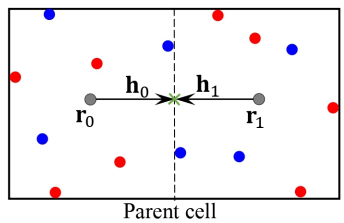

Figure 3: Multipole moments shifting.

The necessary formulae can be derived by calculation the multipole moments with respect to shifted point $(\vec{r}_i + \vec{h}_i)$, expressing the result through the same and lower-order unshifted moments. The resulting shifting rules, where shifting vector $\vec{h}$ is considered as a complex number $z_h = h_x + ih_y$, are the following:

$$m^{(p)}|_{\vec{h}} = \sum_{k=0}^{p} C_p^k m^{(p-k)} z_h^k, \tag{10}$$

where C_p^k is binomial coefficient. It should be noted, that all the presented formulae for the multipole moments and their shifting rules are similar to ones in the original FMM algorithm.

Tensor coefficients $\overline{\overline{\Theta}}^{(p)}$, that initially have been introduced according to eqn (6), now correspond to the complex numbers $\Theta^{(p)}$, and they are determined by only the distance vector between the clusters centers $\vec{\rho}$, represented as the complex number $z_\rho = \rho_x + i\rho_y$. Thus, the formulae, that are the most suitable in practice, have the following form:

$$\Theta^m = \frac{1}{|z_\rho|^2} z_\rho, \quad \Theta^{(p)} = \frac{p}{|z_\rho|^2}\left(\Theta^{(p-1)} \cdot z_\rho\right), \quad p \in \mathbb{N}. \tag{11}$$

In order to calculate complex analogues E_q of the tensor coefficients $\overline{\overline{E}}_q$ arising in local expansions introduced initially by eqn (8), it is necessary to summarize products of $\Theta^{(p)}$ and conjugate numbers $\overline{m}^{(p)}$:

$$\begin{aligned} E_0 &= \Theta^m \cdot m^m + \Theta^d \cdot \overline{m}^d + \frac{1}{2!}\Theta^q \cdot \overline{m}^q + \frac{1}{3!}\Theta^o \cdot \overline{m}^o + \frac{1}{4!}\Theta^h \cdot \overline{m}^h, \\ E_1 &= -\Theta^d \cdot m^m - \Theta^q \cdot \overline{m}^d - \frac{1}{2!}\Theta^o \cdot \overline{m}^q, \\ E_2 &= \Theta^q \cdot m^m + \Theta^o \cdot \overline{m}^d, \\ E_3 &= -\Theta^o \cdot m^m. \end{aligned} \tag{12}$$

If one considers six multipole moments instead of five as we did, it is necessary to introduce additional coefficient E_4 in local expansion and add one more term in each expression for $E_0 \ldots E_4$ in eqn (12), and similarly to higher number of multipole terms. So, all of the presented formulae, expressed through complex numbers allow applying them for arbitrary higher-order multipole moments and local expansions of corresponding terms.

Finally, in order to calculate the approximate velocity, instead of (7) the following formula for local expansions can be used:

$$\vec{V}(\vec{r} + \Delta\vec{\rho}) \approx \frac{\vec{k}}{2\pi} \times (\vec{U}^0 + \vec{U}^1 + \vec{U}^2 + \vec{U}^3), \tag{13}$$

where vectors $\vec{U}^0$, $\vec{U}^1$, $\vec{U}^2$ and $\vec{U}^3$ depend on the vector $\Delta\vec{\rho}$ and correspond to complex numbers U^q calculated as follows:

$$U^q = \frac{1}{q!} E_q \cdot \overline{z}^q_{\Delta\vec{\rho}}, \quad z_{\Delta\vec{\rho}} = \Delta\rho_x + i\Delta\rho_y, \quad q = 0,1,2,3.$$

2.3 Algorithm

The main steps of the suggested tree-based hybrid Barnes–Hut/multipole algorithm are the following:

1. Tree root formation, bounding all the vortex particles.
2. Hierarchical tree structure construction.
3. Calculation of the leaf cells parameters (complex analogues of multipole moments) according to eqn (9).
4. Upward tree traversal and parent-cells parameters calculation by summation shifted according to eqn (10) children multipole moments.
5. For each leaf cell:
   - downward tree traversal: if the current cell is far enough, i.e., the criterion (3) is satisfied, the local expansion coefficients (12) are accumulated; otherwise the downward traversal is continued or, if cell is a leaf, it is stored as a close cell list;
   - direct influence calculation induced by vortex particles, that are contained in close cells (using the Biot–Savart law);
   - summation of the last result with approximate influence calculated using eqn (13).

3 NUMERICAL EXPERIMENT

3.1 Accuracy investigation

The first numerical experiment is performed for the model problem of velocities calculation in square $h \times h$ control cluster induced by vortex particles contained in square $d \times d$ influence cluster placed at the distance $|\vec{\rho}|$ (Fig. 4).

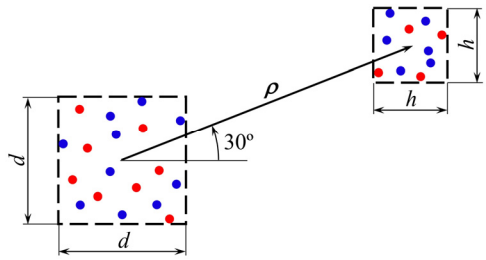

Figure 4: Model problem of two clusters interaction.

In order to estimate the accuracy of the developed method, we assume, that the influence cluster is always placed at the boundary of the close-zone, i.e., $\theta = (2d + 2h)/|\vec{\rho}|$ and we vary it by changing values h and $|\vec{\rho}|$. The relative error is calculated as

$$\delta V = \frac{\frac{1}{N}\sum_{i=1}^{N} |\vec{V}_i^{\,direct} - \vec{V}_i^{\,fast}|}{\max_i |\vec{V}_i^{\,direct}|}, \tag{14}$$

where $\vec{V}_i^{\,direct}$ is the i th vortex velocity calculated directly by using the Biot–Savart law (eqn (2)); $\vec{V}_i^{\,fast}$ – using the proposed Barnes–Hut method modification. Fig. 5 shows the relative error δV dependency against the proximity parameter for the schemes I…IV that correspond to different number of terms accounted in series expansion.

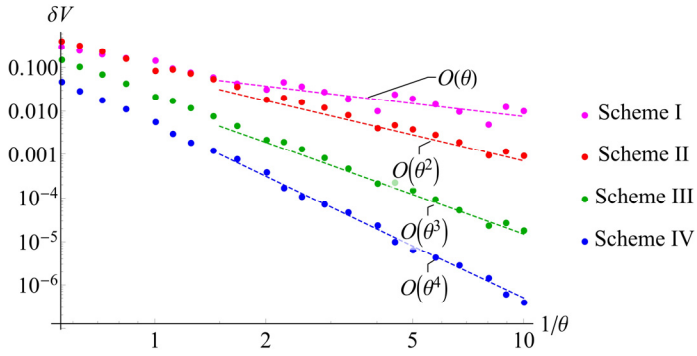

Figure 5: The relative error of the velocities calculation against the inverse proximity parameter.

It is seen that the scheme I provides the first order of accuracy: the error decreases proportionally to θ, while the most accurate scheme IV provides the fourth order of accuracy (with respect to θ).

3.2 Complexity investigation

The method's computational complexity is examined on the problem of velocities calculation, induced by their mutual interaction, for large number of vortex particles: from $5 \cdot 10^4$ to $2 \cdot 10^6$. The number of multiplicative and division operations Q, performed on real numbers, is shown in Fig. 6 for plausible value of proximity criterion $\theta = 0.5$.

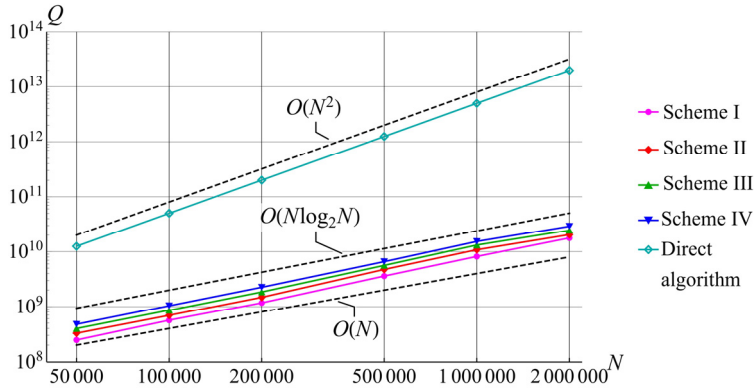

Figure 6: Computational complexities of different fast method schemes against number of vortex particles.

It is seen that computational complexity of all the fast method modifications is quasilinear; the complexity of the most accurate scheme IV is twice as high in comparison to the scheme I, but it is one hundred times more accurate. In comparison to direct particle-to-particle (Biot–Savart) method, that has squared computational complexity, for the most time-consuming problem with 2 millions particles, the suggested fast method provides speedup up to 10^4 times.

3.3 Comparison with the FMM

In order to compare the proposed algorithm with original FMM method, we use in-house implementation of the FMM running on 6-core Intel i7-8700 CPU with OpenMP technology. We compare it to the most accurate version of the proposed fast method (scheme IV) also being run in parallel mode. Firstly, we consider two model problems of mutual interaction calculation for $N = 100000$ and $N = 500000$ vortex particles distributed uniformly in unit square. Calculation time for both methods is shown in Fig. 7, where 'B–H' means the proposed fast method modification. The labels under the 'columns' correspond to the relative error value, which is nearly the same for both methods.

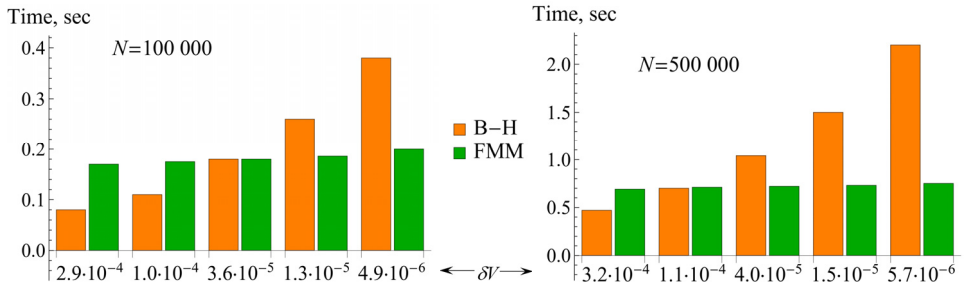

Figure 7: Comparison of the Barnes–Hut method modification with original FMM algorithm for the uniformly distributed vortex particles; $N = 100000$ and $N = 500000$.

It is seen, that the FMM runtime does not change significantly with increasing the accuracy, so the method is especially efficient if the high accuracy is required. At the same time complexity of the proposed fast method depends essentially on the accuracy. It is slightly more efficient than FMM for low accuracy, but its efficiency reduces sharply with increasing both the accuracy and number of vortex particles.

The next numerical experiment is performed for similar problem, but now vortex particles form realistic vortex wake after the bridge cross-section (Fig. 1). Here we also consider two cases with different number of vortex particles – $N = 50000$ and $N = 250000$; calculation time is shown in Fig. 8.

So, that for the typical vortex particles distribution simulating some vortex wake, FMM runtime is two to three times higher in comparison to the proposed Barnes–Hut method modification at medium error level. Thus, for such problems the developed algorithm seems to be obviously preferable.

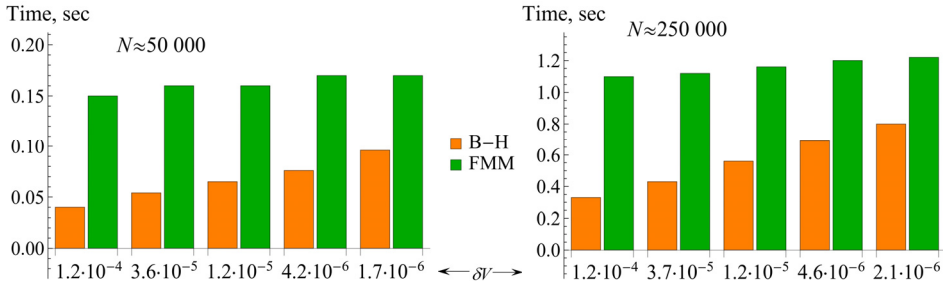

Figure 8: Comparison of the Barnes–Hut method modification with original FMM algorithm for the vortex wake after the bridge section, $N \approx 50000$ and $N \approx 250000$.

4 CONCLUSION

In the paper the Barnes–Hut algorithm modification for vortex particles velocities calculation in 2D vortex method is proposed. This method has quasilinear computational complexity and is based on multipole and local expansions of the influence function. It is tree-based algorithm and provides flexible accuracy adjusting in contrast to the original FMM method. Four schemes of different accuracy are proposed, that are distinguished by number of terms in corresponding expansions. Schemes have 1–4 orders of accuracy with respect to proximity parameter. The necessary formulae are derived for multipole moments calculation, their shifting rules and coefficients of the local expansions. The most suitable way for operations with all the tensors is their representation through complex numbers. For the problems with uniform distribution of vortex particles, the proposed algorithm takes approximately the same time as the FMM method at medium accuracy. For the velocities calculation of the particles, that form vortex wake after the bridge cross section, the proposed method is two to three times faster.

REFERENCES

[1] Cottet, G.H. & Koumoutsakos, P.D., *Vortex Methods*: *Theory and Practice*, 2000.
[2] Mimeau, C. & Mortazavi, I., A review of vortex methods and their applications: From creation to recent advances. *Fluids*, **6**(2), 68, 2021.
[3] Guvernyuk, S.V. & Dynnikova, G.Y., Modeling the flow past an oscillating airfoil by the method of viscous vortex domains. *Fluid Dynamics*, **42**(1), pp. 1–11, 2007.
[4] Kuzmina, K., Marchevsky, I., Soldatova I. & Izmailova, Y., On the scope of Lagrangian vortex methods for two-dimensional flow simulations and the POD technique application for data storing and analyzing. *Entropy*, **23**(1), 118, 2021.
[5] Barnes, J. & Hut, P., A hierarchical $O(n \log n)$ force-calculation algorithm. *Nature*, **324**(4), pp. 446–449, 1986.
[6] Dynnikova, G.Y., Fast technique for solving the N-body problem in flow simulation by vortex methods. *Computational Mathematics and Mathematical Physics*, **49**(8), pp. 1389–1396, 2009.
[7] Karras T., Maximizing parallelism in the construction of BVHs, octrees, and k -d trees. *Proc. Intern. Conf. Eurographics/SIGGRAPH*, pp. 33–37, 2012.
[8] Greengard, L. & Rokhlin, V., A fast algorithm for particle simulations. *Journal of Computational Physics*, **73**(2), pp. 325–348, 1987.

[9] Barnes, J., A modified tree code: Don't laugh; it runs. *Journal of Computational Physics*, **87**(1), pp. 161–170, 1990.
[10] Capuzzo-Dolcetta, R. & Miocchi, P., A comparison between fast multipole algorithm and tree-code to evaluate forces in 3-D. *Journal of Computational Physics*, **143**(1), pp. 28–48, 1997.
[11] Bode, P., Ostriker, J.P. & Xu G., The tree-particle-mesh N-body gravity solver. *Astrophysical Journal Supplementary*, **128**(2), pp. 561–570, 2000.
[12] Hockney, R.W. & Eastwood, J.W., *Computer Simulation using Particles*, CRC, 1988.
[13] Li, P., Johnston, H. & Krasny, R., A Cartesian treecode for screened Coulomb interactions. *Journal of Computational Physics*, **228**, pp. 3858–3868, 2009.