The Barnes — Hut-type algorithm in 2D Lagrangian vortex particle methods

E Ryatina$^{1,2}$ and A Lagno$^1$

$^1$Bauman Moscow State Technical University, 2-nd Baumanskaya st., 5, Moscow 105005, Russia
$^2$Ivannikov Institute for System Programming of the Russian Academy of Sciences, Alexander Solzhenitsyn st., 25, Moscow 109004, Russia

E-mail: evgeniya.ryatina@yandex.ru

Abstract. The Lagrangian vortex particle method for two-dimensional flow simulation around airfoils is considered. Generally, the most time-consuming operation in the vortex method is vortex particles interaction simulation, especially if it is performed directly “point-to-point” for all pairs of particles. The result of such operation is vortex particles convective velocities. This problem is similar to the $N$-body problem and has squared computational complexity $O(N^2)$. The only efficient approach for its solution in a reasonable time for large $N$ is to use approximate fast algorithms. The fast method suggested by Barnes and Hut having logarithmic computational complexity is considered for the mentioned problem of convective velocities computation. It also has been adapted for other operations in the vortex method such as a system of algebraic equations iterative solution and efficient right-hand side computation. As the result, the Barnes — Hut method allows reducing the whole time of calculations by several tens of times.

1. Introduction

The Vortex Particle Method (VPM) [1, 2] is a meshless purely Lagrangian method for flow simulation around airfoils and it is a powerful tool for numerical simulation in some engineering applications connected with hydro- or aerodynamic loads estimation, fluid-structure interaction problems, etc. when the flow compressibility can be neglected. The main idea is to consider the vorticity $\Omega$ as a primary computational variable. So, instead of the classical Navier — Stokes equations with primitive (velocity-pressure) variables in 2D case, we deal with the equation in the following form [3]:

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

(1)

where $V$ is convective flow velocity, $W$ is diffusive one, caused by viscosity effect. Note, that vorticity field $\Omega$ has only one non-zero component, orthogonal to the flow plane, so hereinafter we denote $\Omega = \Omega k$.

The equation (1) can be treated as a transfer equation for the vorticity with velocity $V + W$. New vorticity is generated only on the airfoil surface line. Vorticity distribution in the flow domain is simulated by a set of $N$ elementary vorticity carriers — vortex particles, which are characterized by circulations $\Gamma_i$, positions in the flow domain $r_i$ and small constant radius $\varepsilon$. 
An airfoil in the flow is approximated by a polygon, that consists of \( n \) rectilinear panels. The velocity field can be reconstructed using the generalized Biot — Savart law

\[
V(r,t) = V_\infty + \sum_{i=1}^{N} \Gamma_i \frac{k \times (r - r_i)}{2\pi |r - r_i|^2} + \sum_{i=1}^{n} \int_{K_i} \gamma(\xi,t) \frac{k \times (r - \xi)}{2\pi |r - \xi|^2} d\xi + \int_{K_i} \gamma^{att}(\xi,t) \frac{k \times (r - \xi)}{2\pi |r - \xi|^2} d\xi,
\]

where \( K_i \) is the \( i \)-th airfoil panel; \( n \) is number of panels; \( N \) is number of vortex particles; \( V_\infty \) is incident flow velocity; \( \gamma(r,t) \) is unknown vortex sheet intensity; \( \gamma^{att}(r,t) \) and \( q^{att}(r,t) \) are attached vortex and source sheets intensities, equal to tangent and normal components of the surface line velocity, respectively. For a given law of the airfoil motion the only unknown variable at each time step is \( \gamma(r) \). It can be found from the no-slip boundary condition, equivalent to the equality between tangent velocity components on the airfoil surface line (equality of normal components will be satisfied automatically) [4]. From mathematical point of view, it leads to the boundary Fredholm-type integral equation of the 2-nd kind

\[
\int_{K} Q_\tau(r,\xi)\gamma(\xi) d\xi - \frac{1}{2} \gamma(r) = f(r), \quad Q_\tau(r,\xi) = \frac{k \times (r - \xi)}{2\pi |r - \xi|^2} \cdot \tau(r),
\]

\[
f(r) = -\left( (V_\infty - \frac{1}{2}V_K(r)) \cdot \tau(r) + \sum_{i=1}^{N} Q_\tau(r,r_i)\Gamma_i + \sum_{i=1}^{n} \int_{K_i} Q_\tau(r,\xi)\gamma^{att}(\xi) d\xi + \int_{K_i} q^{att}(\xi)\gamma^{att}(\xi) d\xi \right),
\]

where \( \tau \) is tangent unit vector, chosen such as \( n \times \tau = k \); \( n \) is unit outer normal vector; \( V_K \) is airfoil boundary velocity. The out-of-integral term arises from the jump discontinuity of the velocity field (2) on the airfoil surface line. The kernel of the equation (3) is bounded for smooth airfoils and unbounded, but integrable for airfoils with corner points. To solve this equation numerically, a hierarchy of schemes has been developed for piecewise-constant, piecewise-linear or piecewise-quadratic solution representation [6]. Note, that this mathematical model can be easily generalized for the case of flow simulation around the system of airfoils, both immovable or moving.

Thus, the whole algorithm of the VPM includes several stages at every time step. Firstly, the vorticity generation at the airfoil surface line is modelled. It leads to the above mentioned integral equation approximated, in turn, with a system of linear algebraic equations (SLAE), usually solved by Gaussian elimination method. Then the velocity (convective and diffusive, both being computed through mutual interactions of all vortex particles) computation block is executed. After that, hydrodynamic loads computation (including forces, torque, and pressure field reconstruction) is performed. And finally, vortex particles motion and the restructuring subroutine for vortex wake follow, including merging of closely-spaced vortex particles, vortex particles removing that are far from the airfoil, and no-penetration control for vortex particles. Note, that it is necessary to perform all of these operations at every time step, while the number of such steps usually has an order of few tens thousands.

If a number of vortex particles \( N \) is large enough, the most time-consuming operation is vortex velocities computation (the second term in (2), if it is calculated for all the elements “directly”). This problem is similar to the gravitational \( N \)-body problem, which computational complexity is
proportional to $N^2$ and it can be significantly reduced by using the so-called fast methods having quasilinear computational complexity. We consider the Barnes — Hut method [5], that initially had been developed for the gravitational problem and then was adapted to vortex methods [9]. The method has logarithmic computational complexity $O(N \log N)$ and also can be applied for efficient implementation of the other VPM operations. Taking into account that vortex convective velocities computation is equivalent from a mathematical point of view to multiplying of a dense matrix $N \times N$ by a vector, the fast method also can be applied for iterative SLAE solving. Note, that the right-hand side calculation according to (5) is similar to the convective velocity computation by formula (2) except two simplest terms, so its calculation also can be performed using the fast method. Thus, the Barnes — Hut method implementation for different computational blocks of the VPM algorithm can significantly reduce the computation time of every time-step and whole problem solution.

So, the goal of the research is to implement the Barnes — Hut-type method for vortex velocities computation and to adapt it to other time-consuming computational blocks of the VPM algorithm.

2. The Barnes — Hut-type method

The main idea of this method is that the influence of the groups (clusters) of closely adjacent vortex particles on other such groups located far apart from each other, can be calculated approximately using linearised formulae, derived in [9]. For this purpose, the hierarchical tree-structure of rectangular space domains (cells) is constructed in the flow domain. The zero-level cell contains all the vortex particles, and then it is divided across its long side into two first-level cells. Each of them is reduced horizontally and vertically according to its vortices to exclude empty area. Similarly, the second-level cells are constructed, etc. Such a procedure is performed repeatedly until the target level is achieved or the cell contains single vortex. The whole algorithm consists of the following stages:

(i) Zero-level cell formation which contains all the vortex particles.
(ii) Tree structure construction.
(iii) Calculation of the tree-cells parameters (centres of positive and negative vorticity and total circulations).
(iv) For every terminal tree-cell the following operations are performed:
   • tree traversal and determination of the far-spaced cells according to chosen proximity criteria;
   • accumulation of the linear expansion coefficients for all far-placed cells;
   • exact calculation of the influence from the vortices in cells from neighbouring zone according to the Biot — Savart law;
   • approximate calculation of the contribution of the far cells.

Initially, this method has been developed for the convective velocities computation [9]. Then assuming that the right-hand side of the SLAE has a similar form, this method also has been adapted for its fast computation. In this case “direct” calculation of the neighbouring zone contribution is performed using special formulae which choice should be consistent with the chosen numerical scheme and the airfoil approximation. The contribution of far-placed clusters of vortex particles is replaced with only their vorticity centres influences (positive and negative separately). The error of the method in comparison to the direct calculation is determined only by chosen proximity criteria. We assume that the acceptable error level for vortex method is about 0.2%. Note, that number of tree levels effects only on computational complexity, moreover, for each problem, there is an optimal value of the tree depth that leads to minimal computational complexity.
The other time-consuming VPM operations such as diffusive velocities computation and restructuring subroutine (more precisely, close particles search for their further merging) could be sped up significantly using the above-described tree structure in the flow domain. Both mentioned operations should be performed only in the neighboring zone of each cell, far cells influence can be neglected.

In the numerical experiment, we consider the problem of the flow simulation around the oscillating bridge section (Fig. 1).

![Figure 1. Vortex wake around bridge section (general and detailed views)](image)

The bridge profile is approximated by \( n \approx 2000 \) rectilinear panels. Time statistics of different VPM operations performed using the Barnes — Hut method in comparison to the direct calculation is shown in table 1 for a particular time step with \( N \approx 50000 \) vortex particles in vortex wake.

| Operation                      | Direct method | Fast method | Speedup |
|--------------------------------|---------------|-------------|---------|
| Right-hand side                | 7.08 sec      | 0.10 sec    | 70.80   |
| Convective velocities          | 83.56 sec     | 1.09 sec    | 76.66   |
| Diffusive velocities           | 21.07 sec     | 1.52 sec    | 13.86   |
| Restructuring subroutine       | 23.01 sec     | 1.43 sec    | 16.09   |
| Time step                      | 134.81 sec    | 4.29 sec    | 31.42   |

As a result, the Barnes — Hut method usage for different VPM operations provides 30 times speedup of the time step execution time. The achieved error level for convective and diffusive velocities and the right-hand side is less than 0.2% in comparison to the direct calculation.

3. Boundary integral equation solution

If a system of several airfoils is considered, which somehow move relatively to each other, the problem of efficient SLAE solution appears. Elsewise (if the airfoils are immovable relative to each other) the matrix coefficients remain constant, so it is possible to inverse the matrix at once and then multiply it by the right-hand side vector at every time step. If matrix coefficients are not constant, the Gaussian elimination method can be time-consuming in case of detailed
airfoils discretization. This problem can be solved in some sense by applying iterative algorithms. We consider two different methods: biconjugate gradient stabilized method (BiCGStab) and generalized minimal residual algorithm (GMRES). Another problem is related to memory requirements: the necessity of all the coefficients storage can limit significantly the resolution of a surface mesh on the airfoils.

All iterative methods in the general case involve $N \times N$ matrix multiplication by some vector $[A]\{x\}$. It can be a previous trial, correction vector, residual vector or some other. The matrix $[A]$ coefficients depend on the chosen numerical scheme. In the simplest case vortex sheet intensity over each panel can be shrunk to a point vortex (placed, as a rule, at panel centre). In this case, the matrix coefficients may be considered as interactions between panels centres with corresponding circulations. Imagine, that for some iterative solver we need to multiply matrix $[A]$ by a previous trial $[\Gamma]$. This operation is similar to the vortex velocities computation, due to the kernel $Q(\tau, \xi)$ of the boundary equation (3) is related to the influence function $Q(\tau, \xi)$ as following: $Q^*(\tau, \xi) = Q(\tau, \xi) \cdot \tau(\tau)$. In more accurate schemes the vortex sheet intensity distribution over panels is taken into account [6, 7, 8]. However, the influence of far-placed panels or even clusters of such panels can be calculated approximately as the influence of one point vortex, while interactions between closely placed panels should be calculated directly (exactly) using the chosen numerical scheme. Such a “hybrid” approach for numerical scheme implementation makes it possible to apply the fast Barnes — Hut method for the system iterative solving.

As an example, we consider numerical scheme $\mathcal{T}_{layer}$ as an accurate one and the scheme $\mathcal{T}_{vort}$ as a coarse (“point-to-point” interaction) scheme. These schemes provide for some test problem the relative errors $1 \cdot 10^{-6}$ and $5 \cdot 10^{-4}$, respectively. If we consider $q$ as a ratio of coefficients calculated according to the exact $\mathcal{T}_{layer}$ scheme (the rest are calculated approximately according to the $\mathcal{T}_{vort}$), the resulting error decreases rapidly (Fig. 2), so in practice, it seems to be enough calculating about only 5% of coefficients accurately.

![Figure 2](https://example.com/figure2.png)

**Figure 2.** The “hybrid” scheme error against the ratio $q$ of coefficients calculated accurately

In the general case we deal with some vector $\{x\}$ depending on the iterative solver instead of vortex elements circulations vector, however, it does not affect the idea of fast method usage.
A numerical experiment was performed for the elliptical airfoil with axes ratio 10 : 1 approximated with \( n \) rectilinear panels. Here we consider a model problem to estimate the efficiency of the described approach, so for simplicity, we assume the airfoil to be immovable with no vortex wake in the flow domain. In this case, the right-hand side depends only on the incident flow velocity. Values of SLAE solving computational time using Gaussian elimination and iterative methods (BiCGStab and GMRES) for the mentioned “hybrid” numerical scheme are shown in Fig. 3 in log-log scale, where the asymptotic dependencies \( O(n^3) \), \( O(n^2) \) and \( O(n \log_2 n) \), respectively, are shown by solid lines for large values of \( n \).

![Figure 3. SLAE calculation time for elliptical airfoil approximated by \( n \) panels, solid lines correspond to asymptotic dependencies \( O(n^3) \), \( O(n^2) \) and \( O(n \log_2 n) \)](image)

It is seen that both BiCGStab and GMRES methods provide approximately the same computational time, but the second one is slightly more efficient. The Barnes — Hut method usage leads to error level less than 0.06\%, which we assume acceptably.

So, for the elliptical airfoil with \( n = 10^4 \) panels fast algorithm usage speed up SLAE solution by GMRES method approximately 12 times, while GMRES method itself provides 71 times speedup in comparison to Gaussian elimination method. Note, that such an approach also allows not to store all the matrix coefficients. Here it is necessary to store only close to diagonal matrix elements, calculated exactly.

4. Conclusions
The efficiency of the fast Barnes — Hut method adaptation is investigated for the time-consuming operations in the Vortex Particle Method: convective and diffusive velocities computation, the vortex wake restructuring subroutine, linear equations system right-hand side computation and its iterative solution. For the last operation, the “hybrid” numerical scheme was used for matrix coefficients calculation. The Barnes — Hut method allows for reducing computational time for the mentioned operations significantly, herein the speedup of one particular time step is about 30 times. As a result, the whole calculation time of the problem is reduced by several tens of times, that permits one to extend the class of problems to be solved using VPM.
The idea of the Barnes — Hut method is very simple, and as it is shown in this paper, it can be applied to efficient performing of some other operations in VPM. However, if the number of vortex particles becomes extremely large, the efficiency of the Barnes — Hut method for convective velocities computation, can become lower in comparison to other types of fast methods, such as fast multipole method (FMM) [10] and the Particle-particle particle-mesh (P3M) method based on the fast Fourier transform technique [11]. At the same time, FMM and P3M hardly can be efficiently adopted for diffusive velocities calculation and vortex wake restructuring subroutine.

Acknowledgments
Work of E. Ryatina is supported by Russian Science Foundation (proj. 17-79-20445).

References
[1] Cottet G-H and Koumoutsakos P 2000 Vortex methods. Theory and practice UK: CUP
[2] Lewis R I 1991 Vortex element methods for fluid dynamic analysis of engineering systems UK: CUP
[3] Dynnikova G Ya 2004 The Lagrangian approach to solving the time-dependent Navier — Stokes equations Dokl. Phys. 49 648–52
[4] Kempka S N, Glass M W, Peery J S, Strickland J H and Inger M S 1996 Accuracy considerations for implementing velocity boundary conditions in vorticity formulations Sandia Rep. Sand96-0583 UC-700
[5] Barnes J and Hut P 1986 A hierarchical $O(N \log N)$ force-calculation algorithm Nature 324(4) 446–9
[6] Kuzmina K S and Marchevsky I K 2019 On the calculation of the vortex sheet and point vortices effects at approximate solution of the boundary integral equation in 2D vortex methods of computational hydrodynamics Fluid Dyn. 54 991–1001
[7] Kuzmina K S, Marchevskii I K and Moreva V S 2018 Vortex sheet intensity computation in incompressible flow simulation around an airfoil by using vortex methods Math. Models Comput. Simul. 10 276–87
[8] Kuzmina K S, Marchevskii I K, Moreva V S and Ryatina E P 2017 Numerical scheme of the second order of accuracy for vortex methods for incompressible flow simulation around airfoils Russ. Aeronaut. 60 398–405
[9] Dynnikova G Ya 2009 Fast Technique for solving the N-body problem in flow simulation by vortex methods Comput. Math. Math. Phys. 49 1389–96
[10] Greengard L and Rokhlin V 1987 A fast algorithm for particle simulations J. Comput. Phys. 73 325–48
[11] Morgenthal G and Walther J H 2007 An immersed interface method for the Vortex-In-Cell algorithm Comput. Struct. 85 712–26