ParCFD'2020
32^{nd} International Conference on Parallel Computational Fluid Dynamics
May-11-13 2020, Nice FRANCE

ON THE EFFICIENCY OF FAST METHODS FOR VELOCITY RECONSTRUCTION IN 2D VORTEX PARTICLE METHODS

I.K. MARCHEVSKY[*,†], **K.S. KUZMINA**[*,†], **AND E.P. RYATINA**[*,†]

[*]Bauman Moscow State Technical University
Department of Applied Mathematics
105005, 2-nd Baumanskaya st., 5, Moscow, Russia

[†]Ivannikov Institute for System Programming of the RAS
109004, Alexander Solzhenitsyn st., 25, Moscow, Russia

e-mail: iliamarchevsky@mail.ru

Key words: Vortex Methods, Fast Methods, Barnes — Hut method, Fast Fourier Transform, Parallel Technologies

Abstract. The Lagrangian vortex particle method for two-dimensional flow simulation around airfoils is considered. Generally the most time-consuming operation in vortex method is vortex influence computation (squared computational complexity). This problem is similar to the N-body problem and the efficient approach for its solution in a reasonable time for large N is to use approximate fast algorithms. Two different methods of logarithmic computational complexity are considered: the Barnes — Hut method and method based on the Fast Fourier Transform technique. The parallel implementations of these algorithms are developed which allow to achieve significant speedup and make such approaches much more efficient in comparison with "direct" calculation.

1 INTRODUCTION

The vortex particle method 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, etc., when the flow compressibility can be neglected. The main idea is to consider the vorticity Ω to be a primary computational variable. So, instead of classical Navier — Stokes equations with primitive (velocity-pressure) variables in 2D case we deal with "transfer" equation for the vorticity with velocity which consists of convective (flow velocity) and diffusive (caused by viscosity effect) velocities [1]. The vorticity field is simulated by a set of N elementary vorticity carriers — vortex elements, which are characterized by circulations Γ_i and positions in the flow domain $\vec{r}_i$. So in the simplest case convective velocities can be calculated using Biot — Savart law

$$\vec{V}(\vec{r}) = \int\limits_{S(t)} \Omega(\vec{\xi}) \frac{\vec{k} \times (\vec{r} - \vec{\xi})}{2\pi \left|\vec{r} - \vec{\xi}\right|^2} dS = \sum_{i=1}^{N} \Gamma_i \frac{\vec{k} \times (\vec{r} - \vec{r}_i)}{2\pi \left|\vec{r} - \vec{r}_i\right|^2}. \quad (1)$$

If number of vortex elements N is large, it is one of the most time-consuming operation, which is usually performed for all the elements as "point-to-point". It's computational complexity proportional to N^2, so this problem is similar to the gravitational N-body problem. Its computational complexity can be significantly reduced by using the so-called fast methods having quasilinear computational complexity. We consider two methods having $O(N \log N)$ computational complexity: hierarchical Barnes — Hut method and mesh-based method based on the fast Fourier transform (FFT) technique. Its efficient sequential and parallel implementations are developed.

2 BARNES — HUT METHOD

The Barnes — Hut method initially have been developed for the gravitational N-body problem and later was adapted to vortex methods [3]. It is a hierarchical algorithm based on the binary tree of rectangular spacial domains (cells) in the flow domain. Then influences of all vortex elements located in some terminal cell from another cell, which is far apart from the first one, can be calculated approximately as from a pair of vortices with summary positive and negative circulations, located at vorticity "centres of masses". So, we take into account monopole and dipole moments in the Laurent expansion at computation of the influence from the group of closely placed vortex elements and linear terms in the Taylor expansion for the influence approximate computation at closely placed elements.

The algorithm includes such operations as tree construction; calculation of tree-cells parameters; approximate influence computation by coefficients of linear expansion accumulation using tree transversal; addition of the close zone exact influence. The parallel implementation of these operations except for the first two ones was developed using OpenMP and MPI technologies. For cluster systems with multicore nodes it is possible to use both technologies simultaneously.

3 FFT BASED METHOD

The second method is based on the possibility of the convolution integral (1) calculation using Fast Fourier Transform (FFT) technique. For this purpose the uniform rectangular mesh contains $M \times M$ nodes is introduced. As it is shown in [4], the usage of this approach "directly" leads to the significant error caused by inaccurate calculated influence from vortex elements located in some neighboring zone. So, the special correction procedure is required. The main idea is to exclude the inaccurately calculated influence from the neighboring zone of each cell and add the accurate one, calculated directly using the Biot — Savart law. As was found in numerical experiment, that the optimal neighboring zone size is 3 cell layers [4].

The FFT-based algorithm can be split into 3 blocks: nodal circulations computation by using some smoothing operator and correction velocities calculation (which afterwards should be subtracted); convolution integral calculation using the FFT technique; velocities interpolation from the mesh nodes onto the vortex elements and addition the accurately (exactly) calculated vortex influence from the neighboring zone of each cell. The parallel

implementation of the algorithm was developed using OpenMP, MPI technologies (for first and third mentioned blocks) and Nvidia CUDA technology (for second and third blocks).

4 RESULTS

The achieved accelerations by using OpenMP and MPI technologies for the vortex influence computation for time-consuming problem with $N = 10^6$ vortex elements are shown in Fig. 1. The calculations were performed at shared memory system with 16 cores. The results are in good agreement with Amdahl's law with percent of sequential code: 7 % for Barnes — Hut method; 5 % and 8 % for FFT-based method (MPI and OpenMP implementations respectively).

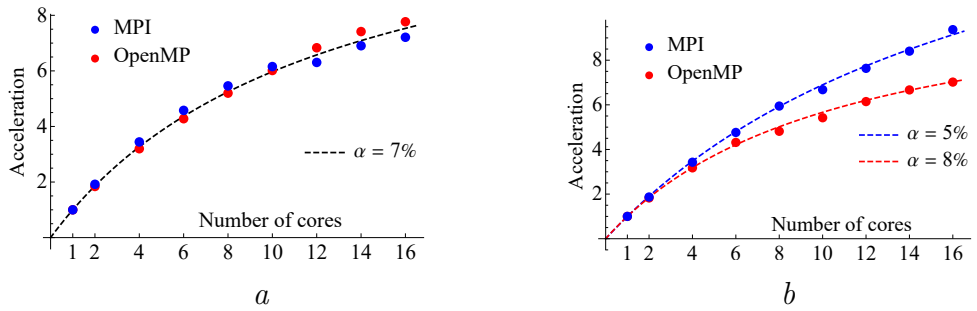

Figure 1: Barnes — Hut (a) and FFT-based (b) methods acceleration on the shared memory system

The comparison with the direct method for the same problem is shown in table 1 where 4 time-consuming problems with different number of vortex elements is considered. The computations were performed in parallel mode on 16-cores CPU and graphics accelerator Tesla V100.

Table 1: Vortex influence calculation time using different methods

	Shared memory system (16 cores)			Graphics accelerator: Tesla V100	
$N \cdot 10^{-5}$	Direct	Barnes – Hut	FFT	Direct	FFT
1	2.80	0.14	0.04	0.05	0.036
2.5	17.58	0.39	0.11	0.33	0.005
5	70.35	0.78	0.20	1.20	0.091
10	280.28	1.68	0.43	4.71	0.170

For the model problem with 10^6 vortex elements the acceleration in comparison with direct method of Barnes — Hut-type algorithm is 167 times, of FFT-based algorithm is 652 times (in parallel mode on 16-cores CPU). The usage of Nvidia CUDA technology for

FFT-based method implementation allows to achieve 27 times acceleration in comparison with direct method (both run on graphics accelerator Tesla V100).

5 CONCLUSIONS

As a result, two fast approximate methods having logarithmic computational complexity for vortex influence computation are considered. Their efficient parallel implementations using OpenMP, MPI and NVidia CUDA technologies are developed. For the model problem with large number of vortex elements ($N = 10^6$) in parallel mode the time of computations can be reduced from 280 seconds (direct method) to 1.68 seconds (the Barnes — Hut method) or even to 0.43 seconds (the FFT-based method). GPU-implementation of the FFT-based method allows to reduce computational time to 0.2 seconds. Note, that all mentioned calculation means the vortex influence computation at one time step. The fast methods implementation for other vortex method operations allows to reduce the whole computational time significantly, which allows us to expand the class of problems to be solved using vortex method. All methods parameters were chosen in such a way that the error level does not exceed 0.2 %.

REFERENCES

[1] Anronov, P.R., Guvernuk S.V. and Dynnikova G.Ya. *Vortex Methods for Computation of Unsteady Hydrodynamic Loads*. Moscow State University Press (2006).

[2] Barnes J. and Hut P. A Hierarchical $O(N \log N)$ Force-Calculation Algorithm, *Nature*. (1986) **324(4)**:446–449.

[3] Dynnikova G.Ya. Fast Technique for Solving the N-body Problem in Flow Simulation by Vortex Methods, *Computational Mathematics and Mathematical Physics*. (2009) **49**:1389–1396.

[4] Morgenthal G. and Walther J.H. An immersed interface method for the Vortex-In-Cell algorithm. *Computers and Structures*. (2007) **85**:712–726.